EURASIP Journal on Applied Signal Processing

Genomic Signal Processing

Guest Editors: Xiaodong Wang, Edward R. Dougherty,
Yidong Chen, and Carsten O. Peterson

Genomic Signal Processing

EURASIP Journal on Applied Signal Processing

Genomic Signal Processing

Guest Editors: Xiaodong Wang, Edward R. Dougherty, Yidong Chen, and Carsten O. Peterson

Contents

EURASIP Journal on Applied Signal Processing 2004:1, 3–4

Editorial

Xiaodong Wang
Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
Email: wangx@ee.columbia.edu

Edward R. Dougherty
Department of Electrical Engineering, Texas A&M University, 3128 TAMU College Station, TX 77843-3128, USA
Email: e-dougherty@tamu.edu

Yidong Chen
National Human Genome Research Institute, National Institutes of Health, Bethesda, MD 20892, USA
Email: yidong@nhgri.nih.gov

Carsten O. Peterson
Department of Theoretical Physics, Lund University, Sölvegatan 14A, SE-22362 Lund, Sweden
Email: carsten@thep.lu.se

The advent of new methods to obtain large-scale surveys of gene expression in which transcript levels can be determined for thousands of genes simultaneously has facilitated the expansion of biological understanding from the analysis of individual genes to the analysis of systems of genes (and proteins). This change characterizes the movement into the era of functional genomics. Central to this movement is an appreciation of the gene's role in cellular activity as it functions in the context of larger molecular networks.

Two salient goals of functional genomics are to screen for key genes and gene combinations that explain specific cellular phenotypes (e.g. disease) on a mechanistic level, and to use genomic signals to classify disease on a molecular level. Signals generated by the genome must be processed to characterize their regulatory effects and their relationship to changes at both the genotypic and phenotypic levels. Since transcriptional (and posttranscriptional) control involves the processing of numerous and different kinds of signals, mathematical and computational methods are required to model the multivariate influences on decision-making in complex genetic networks.

Historically, it has been within the domain of signal processing where such methodologies have been extensively studied and developed—in particular, estimation, classification, pattern recognition, automatic control, information theory, networks, computation, imaging, and coding. Moreover, signal processing is based on a holistic view of regulation and communication. As a discipline, signal processing involves the construction of model systems composed of various mathematical structures, such as systems of differential equations, graphical networks, stochastic functional relations, and simulation models. Therefore it is not surprising that the advent of high-throughput genomic and proteomic technologies is drawing a growing interest from the signal processing community in relation to attacking the fundamental issues of expression-based functional genomics.

The twin aims of tissue classification and pathway modeling require a broad range of signal processing approaches, including signal representation relevant to transcription and system modeling using nonlinear dynamical systems. To capture the complex network of nonlinear information processing based upon multivariate inputs from inside and outside the genome, regulatory models require the kind of nonlinear dynamics studied in signal processing and control. Genomics requires its own model systems, not simply straightforward adaptations of currently formulated models. New systems must capture the specific biological mechanisms of operation and distributed regulation at work within the genome. It is necessary to develop nonlinear dynamical models that adequately represent genomic regulation for diagnosis and therapy.

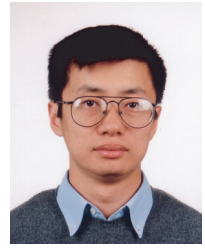
Genomic signal processing (GSP) is the discipline that studies the processing of genomic signals. The aim of GSP is to integrate the theory and methods of signal processing with the global understanding of functional genomics, with special emphasis on genomic regulation. Hence, GSP encompasses various methodologies concerning expression profiles: detection, prediction, classification, control, and

dynamical modelling of gene networks. Moreover, since RNA coding is controlled by DNA sequencing, the analysis of DNA sequences, treated as signals in their own right, can be considered within the domain of GSP. Overall, GSP is a fundamental discipline that brings to genomics the structural model-based analysis and synthesis that form the basis of mathematically rigorous engineering.

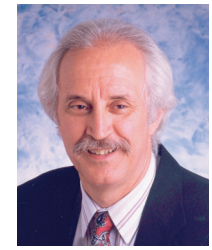
This special issue of EURASIP JASP contains some examples of GSP applications. The issue starts with three papers (Song et al., Chakravarthy et al., and Sussillo et al.) on spectral analysis of DNA sequences. The next paper by Hero et al. treats statistical signal-processing-based gene selection. The following two papers (Wu et al. and Giurcăneanu et al.) develop signal processing techniques for gene clustering. The next two papers treat DNA sequence segmentation using statistical signal processing (Nicorici and Astola) and image processing (Hua et al.), respectively. Signal processing methods for gene prediction and regulatory network inference are developed in the papers by Fox and Carreira, Zhou et al., and Ivanov et al., respectively. The paper by Cristea deals with revealing large-scale chromosome features by analysis of genomic signals. In addition, the paper by Lennartsson and Nordin treats peptides identification using genetic programming. Finally, an invited tutorial by Dougherty et al. discusses key issues in GSP.

The guest editors would like to thank all the authors for contributing their work to this special issue. We would also like to express our deep gratitude to all reviewers for their diligent efforts in evaluating all submitted manuscripts.

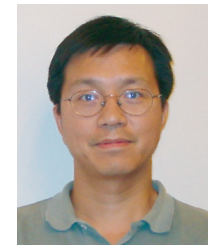
Xiaodong Wang
Edward R. Dougherty
Yidong Chen
Carsten O. Peterson



Xiaodong Wang received the B.S. degree in electrical engineering and applied mathematics (with the highest honor) from Shanghai Jiao Tong University, Shanghai, China, in 1992; the M.S. degree in electrical and computer engineering from Purdue University in 1995; and the Ph.D. degree in electrical engineering from Princeton University in 1998. From July 1998 to December 2001, he was an Assistant Professor in the Department of Electrical Engineering, Texas A&M University. In January 2002, he joined the Department of Electrical Engineering, Columbia University, as an Assistant Professor. Dr. Wang's research interests fall in the general areas of computing, signal processing, and communications. He has worked in the areas of digital communications, digital signal processing, parallel and distributed computing, nanoelectronics, and bioinformatics, and has published extensively in these areas. His current research interests include wireless communications, Monte Carlo based statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an Associate Editor for the IEEE Transactions on Communications, the IEEE Transactions on Wireless Communications, the IEEE Transactions on Signal Processing, and the IEEE Transactions on Information Theory.



Edward R. Dougherty is a Professor in the Department of Electrical Engineering at Texas A&M University in College Station. He holds an M.S. degree in computer science from Stevens Institute of Technology in 1986 and a Ph.D. degree in mathematics from Rutgers University in 1974. He is the author of eleven books and the editor of other four books. He has published more than one hundred journal papers, is an SPIE Fellow, and has served as an Editor of the Journal of Electronic Imaging for six years. He is currently Chair of the SIAM Activity Group on Imaging Science. Prof. Dougherty has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His current research focuses on genomic signal processing, with the central goal being to model genomic regulatory mechanisms. He is Head of the Genomic Signal Processing Laboratory at Texas A&M University.



Yidong Chen received his B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1983 and 1986, respectively, and his Ph.D. degree in imaging science from Rochester Institute of Technology, Rochester, NY, in 1995. From 1986 to 1988, he joined the Department of Electronic Engineering of Fudan University as an Assistant Professor. From 1988 to 1989, he was a Visiting Scholar in the Department of Computer Engineering, Rochester Institute of Technology. From 1995 to 1996, he joined Hewlett Packard Company as a Research Engineer, specialized in digital halftoning and color image processing. Currently, he is a Staff Scientist in the Cancer Genetics Branch of National Human Genome Research Institute, National Institutes of Health, Bethesda, Md, specialized in cDNA microarray bioinformatics and gene expression data analysis. His research interests include statistical data visualization, analysis and management, microarray bioinformatics, genomic signal processing, genetic network modeling, and biomedical image processing.

Carsten O. Peterson is a Professor at the Department of Theoretical Physics and Head of the Complex Systems Division at Lund University, Sweden. His current research area is computational biology with the focus on microarray analysis, genetic networks, systems biology, and alignment algorithms. Dr. Peterson's research interests were initially in theoretical particle physics, multiparticle production, quantum chromodynamics, and also statistical mechanics. His research areas have subsequently evolved into spin systems, data mining, and time series analysis with some emphasis on biomedical applications, resource allocation problems, Monte Carlo sampling methods and mean field approximations, thermodynamics of macromolecules, protein folding/design, and computational biology in general. Dr. Peterson joined the Department of Theoretical Physics at Lund University in 1982, had an industrial intermission with Microelectronics and Computer Corporation (Austin, Tex) during 1986–1988, and held postdoctoral positions at Stanford (1980–1982) and Copenhagen (1978–1979). Dr. Peterson received his Ph.D. degree in theoretical physics and M.S. degree in physics from Lund University in 1977 and 1972, respectively.

EURASIP Journal on Applied Signal Processing 2004:1, 5–12

Comparative Genomics via Wavelet Analysis for Closely Related Bacteria

Jiuzhou Song
Department of Biochemistry and Molecular Biology, Faculty of Medicine, University of Calgary, 3330 Hospital Drive NW, Calgary, Alberta, Canada T2N 4N1
Email: songj@ucalgary.ca

Tony Ware
Department of Biochemistry and Molecular Biology, Faculty of Medicine, University of Calgary, 3330 Hospital Drive NW, Calgary, Alberta, Canada T2N 4N1
Email: tware@ucalgary.ca

Shu-Lin Liu
Department of Biochemistry and Molecular Biology, Faculty of Medicine, University of Calgary, 3330 Hospital Drive NW, Calgary, Alberta, Canada T2N 4N1
Email: slliu@ucalgary.ca

M. Surette
Department of Biochemistry and Molecular Biology, Faculty of Medicine, University of Calgary, 3330 Hospital Drive NW, Calgary, Alberta, Canada T2N 4N1
Email: surette@ucalgary.ca

Received 26 February 2003; Revised 11 September 2003

Comparative genomics has been a valuable method for extracting and extrapolating genome information among closely related bacteria. The efficiency of the traditional methods is extremely influenced by the software method used. To overcome the problem here, we propose using wavelet analysis to perform comparative genomics. First, global comparison using wavelet analysis gives the difference at a quantitative level. Then local comparison using keto-excess or purine-excess plots shows precise positions of inversions, translocations, and horizontally transferred DNA fragments. We firstly found that the level of energy spectra difference is related to the similarity of bacteria strains; it could be a quantitative index to describe the similarities of genomes. The strategy is described in detail by comparisons of closely related strains: *S.typhi* CT18, *S.typhi* Ty2, *S.typhimurium* LT2, *H.pylori* 26695, and *H.pylori* J99.

Keywords and phrases: comparative genomics, gene discovery, wavelet analysis, bacterial genome.

1. INTRODUCTION

Since the publication of the whole genomic sequence of *Haemophilus influenzae* [1], the draft genomes of more than 90 bacterial strains have been completely finished. A notable outcome of these genome projects is that at least one third of the genes encoded in each genome have no known or predictable functions. The genome sequencing, while not providing the detailed minutiae of the complete sequences, allows comparisons between genomes to identify insertion, deletion, and transfers that are undoubtedly important in the different phenotype of strains. However, as the level of evolutionary conservation of microbial proteins is rather uniform, a large portion of gene products from each of the sequenced genomes has homologs in distant genomes [2]. The functions of many of these genes may be predicted by comparing the newly sequenced genomes with those of better-studied organisms. This makes comparative genomics a very powerful approach to a better understanding of the genomes and biology of the organisms and to determine what is common and what unique between different species at the genome level, especially on genome analysis and annotation. In addition, prediction of protein functions, transfer of functional information of paralogs (products of gene duplications) and orthologs (direct evolutionary counterparts), phylogenetic pattern, examination of gene (domain) fusions, analysis of conserved gene strings (operons), and reconstruction of metabolic pathways are facilitated using comparative genomics.

The large amount of data has already given rise to several studies on whole genome comparisons such as those between several closely related bacterial species [3, 4]. One problem for this kind of research is that DNA and protein fragment comparisons are highly dependent on sequence alignment methods such as FASTA34, BLAST, CLUSTALW, STADEN, PHRED, and so forth. Since the efficiency of the methods is extremely influenced by the software methods used, sequence alignment is possible for short DNA and protein sequence comparisons, the methods also need heavy use of time, energy, and resources. Here we propose a strategy for whole genome or large fragment sequence comparisons. The comparative genomics method we propose is based on the whole genome. Firstly, we use wavelet transform analysis to make a global comparison of closely related strains, giving their similarities and differences at quantitative level and with statistical meaning. Then we use keto excess or purine excess, as proposed by Freeman [5], to visualize some local differences. These indices are not like GC skew and AT skew [6, 7, 8] which depend on the sliding window size; they can show the exact positions of rearrangements and the origin and terminus sites of DNA replication. We illustrate the strategy using several closely related species including *S.Typhimurium* LT2, *S.Typhi* CT18, *S.Typhi* Ty2, *H.pylori* J99 and *H.pylori* 26695 strains. These pairs of bacteria share a similar flask-like morphology and show serological cross-reaction, but they differ in several important features including differences in G + C content and genome size, different tissue specificity, and pathogenic effects for human.

To understand the similarity between DNA structure and function, it is necessary to compare DNA sequences, especially for newly closely identified ones. Wavelet analysis has been applied to a large variety of biomedical signals; the method will provide a useful visual description of the inherent structure underlying DNA sequence [9]. A wavelet is a waveform of effectively limited duration that has an average value of zeros, and wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet [10]. It provides a multiscale representation of signals allowing efficient smoothing and/or extraction of basic components at different scales. So the wavelet analysis supplies a new way to compare whole genomes at quantitative levels. The main idea of wavelet analysis is to decompose a sequence profile into several groups of coefficients, each group containing information about features of the profile at a scale of sequence length. Coefficients at coarse scales capture gross and global features, whereas coefficients at fine scales contain the local details of the profile [11]. A wavelet variance is a decomposition of the variance of a signal; it replaces global variability with variability over scales and investigates the effects of constraints acting at different time or space scales [9]. The similarity comparison via wavelet analysis expands the traditional sequence similarity concept, which takes into account only the local pairwise DNA or amino acid sequences and disregards the information contained in coarse spatial resolution. Also the wavelet analysis does not require the complex sequence alignment processing for sequence [12]. In this study, we explore the possibility of genome comparisons using wavelet transform analysis and keto-excess or purine-excess plots to perform comparative genomics, and introduce the idea of using the energy spectra difference as a quantitative index to describe the similarity of genomes. The strategy used in this paper not only provides the location of *oriC* and *terC* sites of DNA replication, but also is a powerful tool for examining genome fragment insertion, inversion, translocation, reorganization, and revealing evolutionary history.

2. MATERIAL AND METHOD

The sequences of *Salmonella typhi* Ty2 [13], *Helicobacter pylori* J99 [14], and *Helicobacter pylori* 26695 [15] were obtained from the NCBI website; *Salmonella typhimurium* LT2 and *Salmonella typhi* CT18 were downloaded from both ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Salmonella_typhimurium_LT2/ and from ftp://ftp.sanger.ac.uk/pub/pathogens/st/, respectively.

For global comparisons of closely related bacteria, we firstly do not use sequence alignment to do the comparison, but use wavelet analysis to compare the purine-excess curve or keto-excess curve [5] and get the genome difference at quantitative level. In transforming the sequence data into digital data, we just count the cumulative number of each of the DNA bases A, C, G, and T along the whole genome. The purine excess was defined as the sum of all purines (A and G) minus the sum of all pyrimidines (T and C) encountered in a walk along the sequence up to the point plotted and was determined by

$$\text{PurineExcess}_n = \left(\sum_{i=1}^{n} B_{\text{A},i} + \sum_{i=1}^{n} B_{\text{G},i} - \sum_{i=1}^{n} B_{\text{T},i} - \sum_{i=1}^{n} B_{\text{C},i} \right), \quad (1)$$

where n ranges from 1 to N (N is the chromosome length) and $B_{\text{A},i}$ is 1 if there is an A in the ith position, and 0 otherwise (the terms $B_{\text{T},i}$, $B_{\text{G},i}$, and $B_{\text{C},i}$ are defined similarly). In the same way, the keto excess was defined as the sum of all keto bases (G and T) minus that of the amino bases (A and C) and was determined by

$$\text{KetoExcess}_n = \left(\sum_{i=1}^{n} B_{\text{T},i} + \sum_{i=1}^{n} B_{\text{G},i} - \sum_{i=1}^{n} B_{\text{A},i} - \sum_{i=1}^{n} B_{\text{C},i} \right). \quad (2)$$

Here again n ranges from 1 to N, where N is the chromosome length, and B is the number of the particular base (A, C, G, or T) occurring at the ith location (either 0 or 1 in each case). We can also define local versions of these vectors:

$$\begin{aligned} KT_n &= B_{\text{T},i} + B_{\text{G},i} - B_{\text{A},i} - B_{\text{C},i}, \\ PT_n &= B_{\text{A},i} + B_{\text{G},i} - B_{\text{T},i} - B_{\text{C},i}. \end{aligned} \quad (3)$$

The fundamental idea behind wavelet analysis is to analyze according to scale [16]. Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions becoming a common

tool for analyzing localized variations of power within a time series, with successful applications in signal and image processing, numerical analysis, and statistics. The wavelet analysis procedure is to adopt a wavelet prototype function called an analyzing wavelet or mother wavelet. Because the original function can be represented in terms of a wavelet expansion (using coefficients in a linear combination of the wavelet function), data operations can be performed using corresponding wavelet coefficients. We employ the continuous real wavelet transform [17]. Our analyzing wavelet is the normalized first derivative of a Gaussian function:

$$\Phi(t) = \frac{t\sqrt{2}}{\pi^{1/4}\sigma\sqrt{\sigma}} \exp\left(-\frac{t^2}{\sigma^2}\right), \tag{4}$$

where σ is a scaling factor. The real wavelet transform of a function f is

$$Wf(t,s) = \int_{-\infty}^{\infty} f(u)\frac{1}{\sqrt{s}}\Phi\left(\frac{u-t}{s}\right)du. \tag{5}$$

In order to apply this transform to a vector $\underline{x}$ of length N (such as the vectors KT or PT defined above), $\underline{x}$ is taken to correspond to samples at the points $t_0 = 0$, $t_1 = 1/N$, $t = 2/N, \ldots, t_N = 1 - 1/N$ of a 1-periodic function $x(t)$. The wavelet transform Wx, for each scale s in a given range, is then just a convolution of two vectors that can be calculated in the Fourier domain using the fast Fourier transform. Explicitly, we have

$$Wx(t_i, s_j) = \sum_n x_n p_{n-i}(s_j), \tag{6}$$

where $p_i(s) = (1/\sqrt{s})\Phi(t_i/s)$, and where the sum is taken over all values n for which the terms in the sum are not negligible. The result is a two-dimensional array of values of Wx at positions t (ranging from 0 to 1) and scales s (a magnification parameter). One can think of this as a collection of one-dimensional transforms of the original signal at different scales.

Methods based on wavelet transforms generally require powerful visualization tools. In implementation, we figure out the purine excess and keto excess using Perl and C++ codes, perform wavelet transformation analysis via Matlab, and make graphics using the xmgrace graphic software on MACI-cluster parallel computers.

3. RESULTS AND ANALYSIS

3.1. Global comparison of the closely related strains

To investigate the relationship between closely related strains and determine their similarity, we use wavelet analysis to show the global spectrum of the two closely related strains. If the spectra are completely identical, they are the same strains, otherwise, we divide them to different strains. This identification, which is different from clone morphological index and physiology and biochemistry characteristics, is based on whole genome comparison. The global wavelet

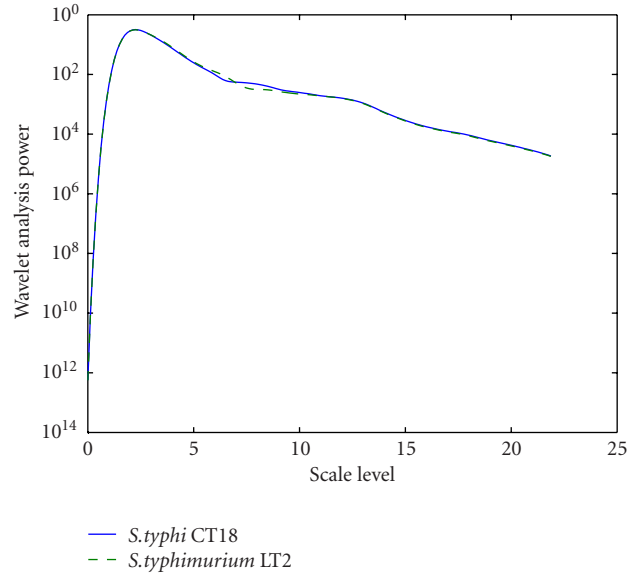


Figure 1: Comparison of the purine-excess wavelet analysis spectra in *S.typhi* CT18 and *S.typhimurium* LT2.

spectra of the purine excess for three pairs of *S.typhi* CT18 and *S.typhimurium* LT2, *S.typhi* CT18 and *S.typhi* Ty2, and *H.pylori* 26695 and *H.pylori* J99 are shown in Figures 1, 2, and 3. The power in the wavelet transform is computed for a range of scales and plotted as a function of scale level σ, where the scale is $s = 2^{-\sigma}$. The higher the scale number is, the shorter the support of the wavelet is, and so the shorter the moving window over which the signal is being measured. From Figure 1, notice the higher energy in the *S.typhi* CT18 starting at scale number 5, corresponding to a length scale of the order of 1/20 of the signal length. Using these wavelet spectra to measure the difference (in a least square sense), we find that the difference between two genomes is of the order of 1.5% of the total signal energy; the quantitative variability is also indicative of component differences in the DNA sequence. This extra variability can be observed in the cumulative signal plots for *S.typhi* CT18, in particular, in the additional features present in the signal (as compared to the corresponding graph for *S.typhimurium* LT2). From Figure 2, the lower energy in another closely-related strains *S.typhi* CT18 and *S.typhi* Ty2 energy spectra, a length scale of the order of 1/20 of the signal length, could be seen. We found that the difference between the two genomes is of the order of 0.7% difference of the total signal energy; it is definitely smaller than that between *S.typhi* CT18 and *S.typhimurium* LT2, which indicates that the similarity between *S.typhi* CT18 and *S.typhi* Ty2 is larger than that of between *S.typhi* CT18 and *S.typhimurium* LT2. From Figure 3, with a same length scale of the order of 1/20 of the signal length, the wavelet spectra measured the difference between *H.pylori* 26695 and *H.pylori* J99; the difference between the two closely related strain genomes is of the order of 17.6% of the total signal energy; it is the biggest difference in the three

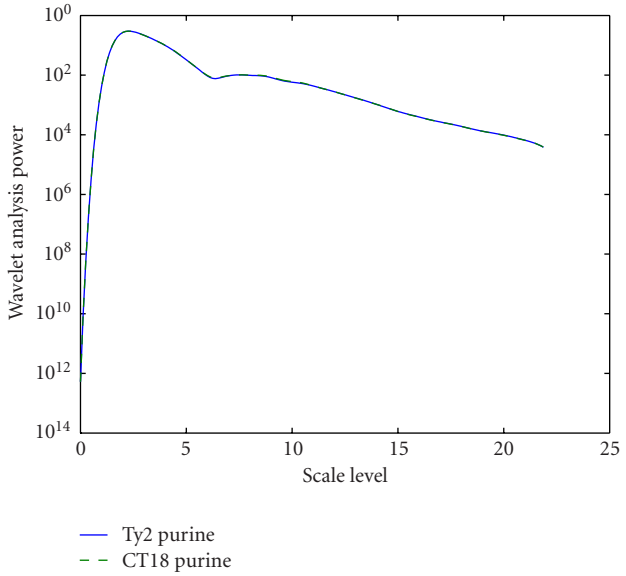


Figure 2: Comparison of the purine-excess wavelet analysis spectra in *S.typhi* CT18 and *S.typhi* Ty2.

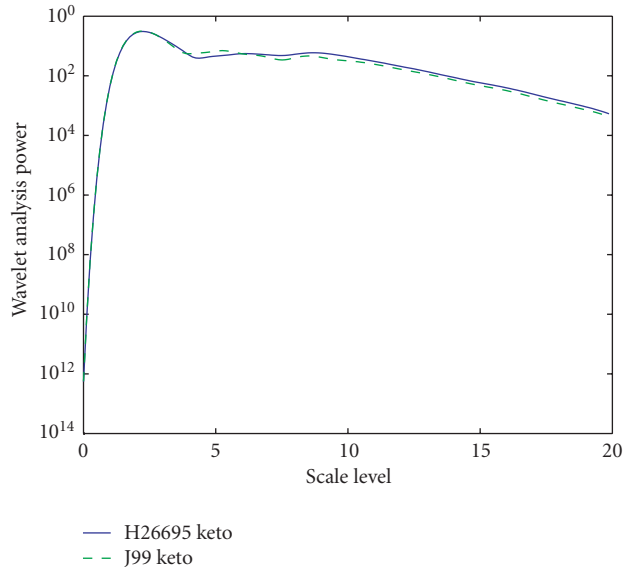


Figure 3: Comparison of the keto-excess wavelet analysis spectra in *H.pylori* 26695 and *H.pylori* J99.

compared closely related strains. Here, we can see that the variability can be observed in the cumulative signal plots for the two strains; the variability is a definite indicative of component differences in the DNA sequences. From the comparisons of the energy spectra among the strains, we can infer that the *S.typhi* CT18, compared to *S.typhimurium* LT2, has closer relationship with and bigger similarity to *S.typhi* Ty2. The strain *H.pylori* 26695 and *H.pylori* J99 have the biggest difference variability in these three compared strains.

3.2. Local comparison of the closely related strains

After comparison via wavelet transformation analysis, we have measured the global difference at a quantitative level. Now we analyze the local differences using the visualized keto-excess or purine-excess plot which explores the main information variation given by the wavelet analysis. In comparative genomics, as shown in Figure 4, the figure clearly shows the positions of terC sites and oriC sites for both strains. Most parts of the keto-excess curves overlap between *S.typhimurium* LT2 and *S.typhi* CT18, but there is an extra part around the terC site in *S.typhimurium* LT2. After partitioning in detail the fragment, the extra fragments in *S.typhimurium* LT2, the fragments A, B, C, D, E, and F in a length range from 1483934 to 1870353 bp as shown in Figure 5a, are rearranged or incompletely translocated to *S.typhi* CT18 which are also located around the terC site; the fragments are completely reversed at the length range from 1235888 to 1643129 bp and the order of fragments is reversed from fragments F to fragment A, as shown Figure 5b. The rearrangements of DNA fragments suggest that the inversions and translocations took place in the strain *S.typhi* CT18 sequences, thus disrupting the original arrangement of these fragments. As a result, the keto excess plot in the *S.typhi* CT18 is a little bit different from that of *S.typhimurium* LT2. As for the transferred or relocated genes, the most inverted fragments in *S.typhi* CT18 involve genes in *S.typhimurium* LT2 which contain cell processes: macromolecule metabolism, cell envelope, energy metabolism, such as secretion system effectors and apparatus [ssa(A–U) and yscR gene], cytoplasmic protein, inner membrane protein, family transport protein, oxidoreductase, periplasmic protein, peptide transport protein, transcriptional regulator or repression, fumarate hydratase, and tyrosine tRNA synthetase. The translocation genes in CT18 include transcriptional regulator, ATPase and phosphatase, ABC superfamily oligopeptide transport protein, peptide transport protein, anthranilate synthase, cardiolipin synthase, energy transducer, formyltetrahydrofolate hydrolase, GTP cyclohydrolase, nitrate reductase, phage shock protein, tryptophan synthase, and so forth.

Another obvious difference of the keto-excess plots in the two closely related strains is that there is a triangle peak around 4.45 mb in *S.typhi* CT18. We noted that Liu (1995) and others found that there was an insertion of length 130 kb in this region in *S.typhi* CT18. From the Keto-excess plot in Figure 4, the insertion of a large DNA fragment is confirmed. After the detailed comparison between *S.typhi* CT18 and *S.typhimurium* LT2 genomes, the insertion of a 35 kb DNA fragment ranging from 44724722 to 4507789 bp was identified in *S.typhi* CT18. DNA fragments G and H in *S.typhi* (Figure 5b) were found to be translocations from *S.typhimurium* LT2, where the fragments range from 2844714 to 2879233 bp (shown in Figure 5a). The translocation genes include regulators of late gene expression, phage

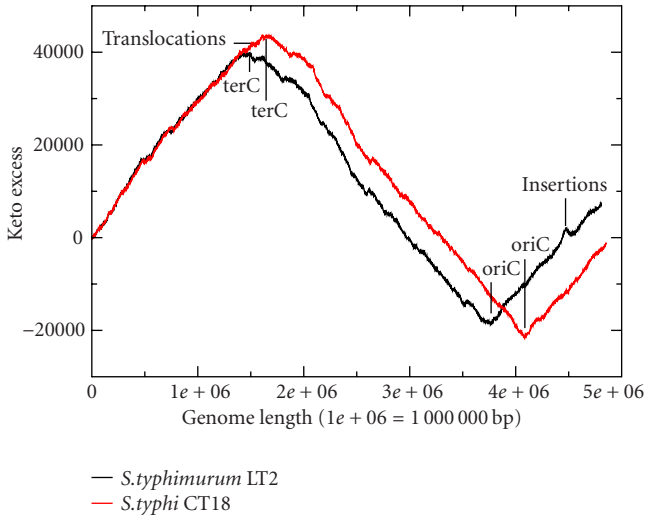


FIGURE 4: Comparative genomics between *S.typhi* CT18 and *S.typhimurium* LT2. The black line is keto-excess plot in *S.typhimurium* LT2 and the red one is keto-excess plot in *S.typhi* CT18. The maximum value and minimum value in each curve are corresponding to the positions of terC site and oriC site of DNA replications, respectively. Compared with *S.typhi* CT18, *S.typhimurium* LT2 has an extra part around terC site; *S.typhi* CT18 has a triangle insertion around 4.45 mb.

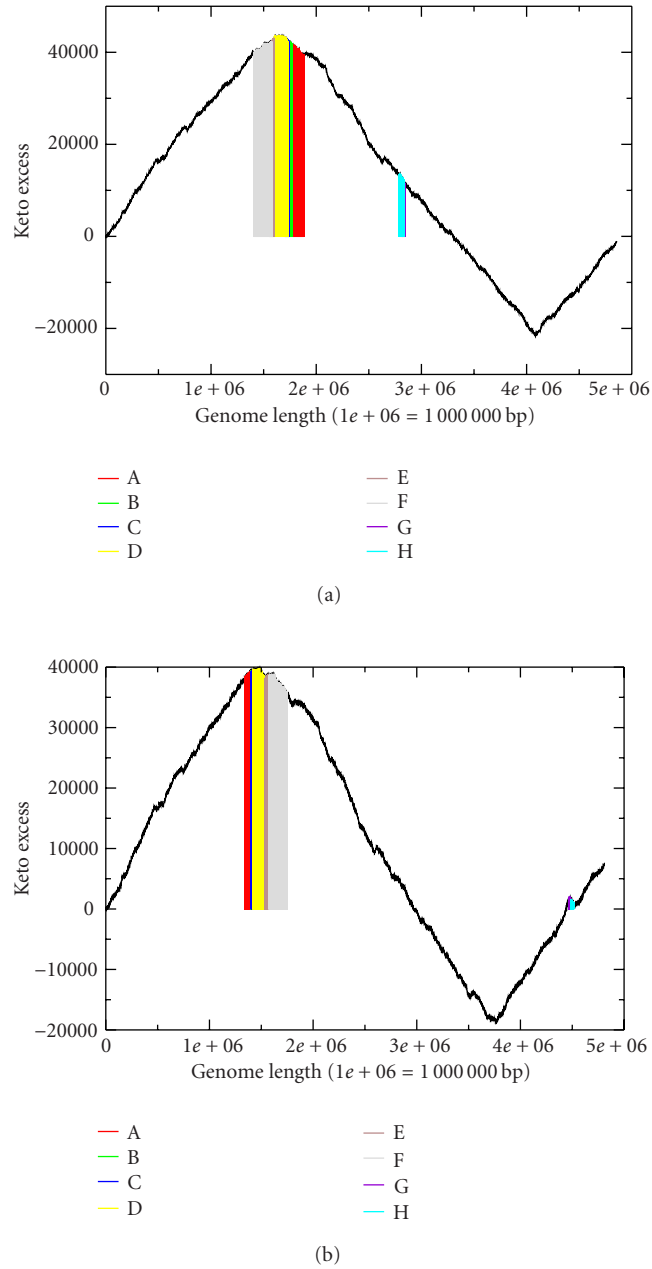


FIGURE 5: Identification of translocated and inserted fragments in *S.typhi* CT18 and *S.typhimurium* LT2. The fragments A, B, C, C, D, E, and F in *S.typhimurium* LT2 are reversed and translocated into *S.typhi* CT18; the order of fragments becomes F, E, D, C, B, A. The partial insertions in *S.typhi* CT18, fragments G and H, are horizontal transferred fragments from *S.typhimurium* LT2; the fragment length of G and H is around 35 KB.

tail protein, phage tail fiber protein, phage base plate assembly protein, lysozyme, membrane protein, and other proteins. The remaining genes within this insertion in *S.typhi* CT18 have not yet been identified.

The numbers and types of paralogs were very different between *S.typhi* CT18 and *S.typhimurium* LT2; those differences also contribute to the local differences of the wavelet transformation spectra and the keto excess-plots in the two strains. In *S.typhimurium* LT2, most of paralogs are two copies of cytochrome c-type biogenesis protein genes (ccmA-H), citrate lyase synthetase (citC-citG), and five copies of transposase (tnpA). In contrast, in *S.typhi* CT18, there are twenty-six copies of transposase (tnpA); the two copies of paralogs are oxaloacetate decarboxylase (oadA, oadB, oadG, and oadX), cytochrome c-type biogenesis protein (ccmA-H), and citrate lyase synthetase (citA-G, X, and T).

The *Salmonella enterica serovar typhi* is a human-specific pathogen causing enteric typhoid fever, a severe infection of the reiculoendothelial system. The *S.typhi* CT18 and *S.typhi* Ty2 are two well-studied pathogenic strains, by the comparison via wavelet spectra they have very little difference and are very close; this statement confirms most of researcher's inference. The information from comparative genomics and genes in *S.typhi* will help us to reveal more specific drug candidates and vaccines. Figure 6 only shows the fragments with larger than 12,000 bp. From Figure 6, the *S.typhi* Ty2 genome is distinguished from that of *S.typhi* CT18 by inter-replichore inversion and translocations. The figure indicates that the inverted DNA fragments are the main reason for the difference between the two strains. There are also a lot of small inverted regions: translocated regions and unique regions (these are not shown here). Through the comparison between the strains, we found besides these major inversions that the gene structures of the two strains are very similar.

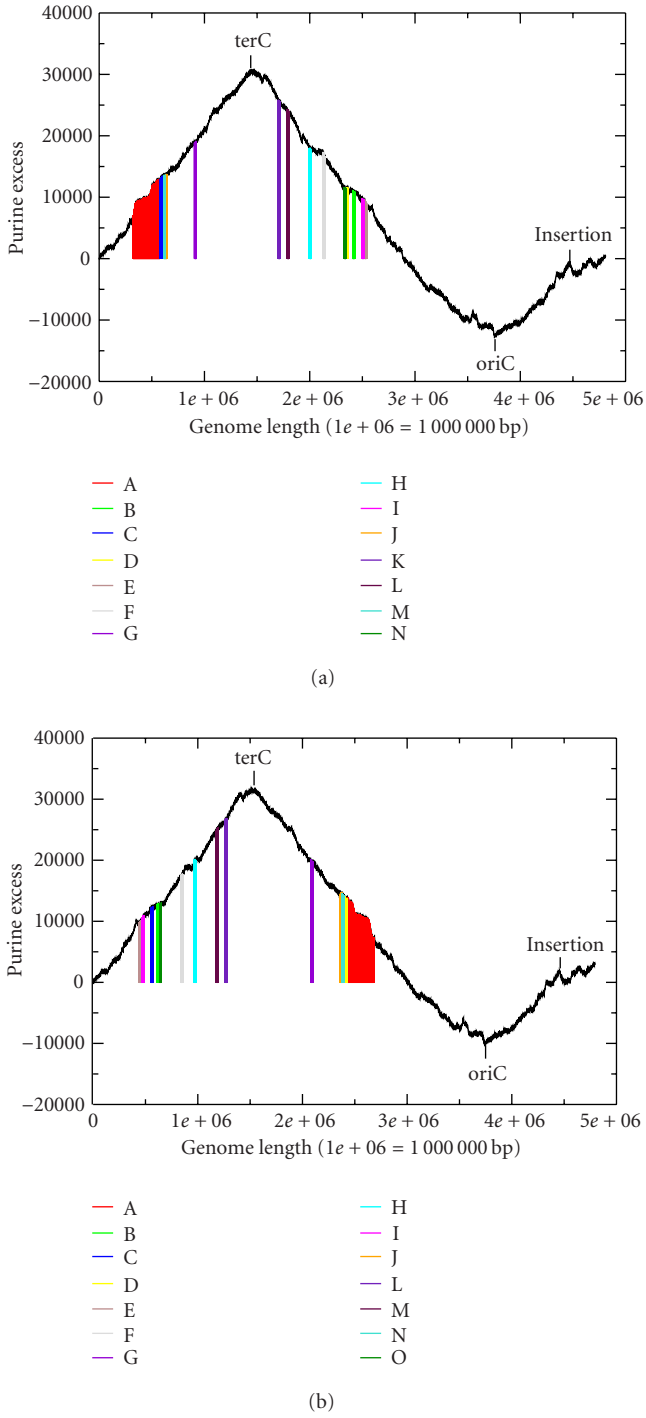


FIGURE 6: Identification of translocated and inserted fragments in *S.typhi* CT18(A) and *S.typhi* Ty2(B). The 14 biggest fragments A, B, C, ..., O in *S.typhi* Ty2 are reversed and translocated into *S.typhi* CT18; the order of fragments becomes O, N, M, ..., A. The partial insertions in *S.typhi* CT18 are horizontal transferred fragments into *S.typhi* Ty2; the fragment length of G and H is around 35 KB.

They have the same positions of oriC and terC site and physical balance features, and share a 35 kb inversions around 4.5 mb. The sequence in the inversion fragment in the two strains is the same as in the fragments G and H of the LT2. We also got a lot of pseudogenes; we think that the inverted and translocated fragments are the main reason of making the pseudogenes in the two strains. The message helps to reveal the pseudogene mechanisms and potentially contributions to pathogenicity; the detail description is beyond the scope of the paper.

Comparative genomics using purine-excess plots was also used to compare *H.pylori* strains J99 and *H.pylori* strains 26695. The size of the inversed and translocated fragments is much smaller than that of *S.typhi* CT18, *S.typhi* Ty2, and *S.typhimurium* LT2, the only fragments larger than 1000 bp are shown in Figure 7. From Figures 7a and 7b, the two strains could clearly show terC sites on the purine curves. We found that the dnaA gene is near the global minimum site, so we refer to the oriC site located on these regions. There are a lot of rearrangements, horizontal transfers, translocations, and reversions among *H.pylori* J99 and *H.pylori* 26695; the inversions and horizontally transformed DNA fragments are clearly seen to result in mirror symmetry transformations. In contrast to previous genomics comparison between the two strains, using window-sized GC skew [18], the purine-excess plots give us precise positions of inversion, translocations, and horizontal transformed DNA fragments. Interestingly, the shape and composition of cag pathogenicity island (cagPAI) are pretty similar. The inversion and translocation events do not happen in this region; this implies that the zone is not a result of differential retention of ancestral DNA in these strains but is a product of horizontal transfer; this region might represent pathogenicity islands [14]. We also found that one of the reasons which formed the jagged diagram of *H.pylori* is that *H.pylori* 26695 has some unique prologs (products of gene duplications). These prologs are acyl carrier protein (acpP), biopolymer transport protein (exbB and exbD), iron dicitrate transport protein (fecA), and transposes (tnpA and tnpB).

4. DISCUSSION

Here we have described a wavelet analysis strategy to reveal the whole genome difference between closely related bacterial strains. Compared with the widely used GC skew and AT skew, the purine excess and the keto excess are visualization tools to show whole genome information; they do not involve any default window size or the loss of any information. Via analyzing the excesses, the wavelet method enables global comparison at a quantitative level, and the keto-excess or purine-excess plot shows the local difference. Through our research, the wavelet energy spectra difference can give a quantitative measure of strain difference. It is an important value for closely related strain, especially for the similar clone morphology and serological cross reaction putative strains. It could be a quantitative index to ascertain the similarity and relationship among strains.

It is worth noting that although we can generate an enormous amount of useful information about the differences

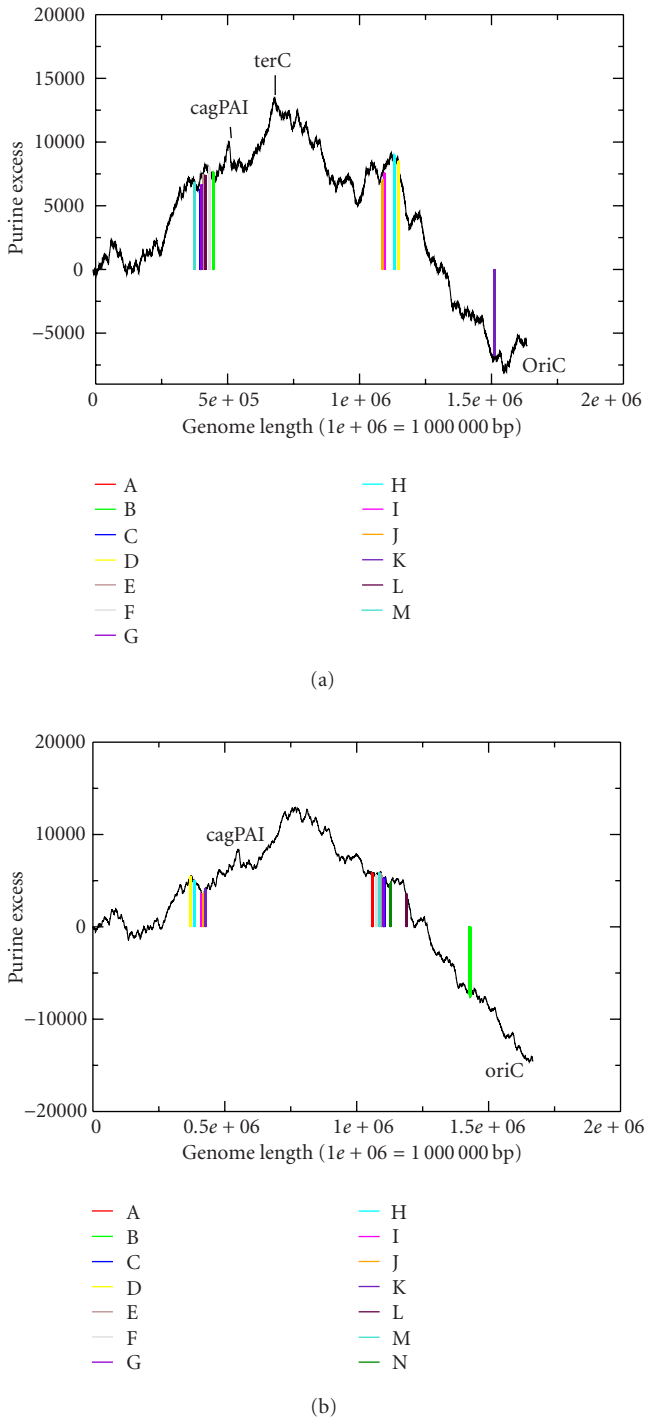


FIGURE 7: Identification of translocated and inserted fragments in *H.pylori*, Strain J99 and *H.pylori*, Strain 26695. The fragments A, B, C, D, E and F in *H.pylori Strain* J99 are reversed and translocated into *H.pylori Strain* H26695.

between closely related strains or species, there is more about comparative genomic analysis other than merely identifying the presence or absence of specific fragments or genes. It is important to know whether these genes are capable of being translated into functional proteins. Very small changes such as insertion, deletion, mutation, translocations, and so forth in genomic sequence can have a disproportionate effects on the phenotype of an organism. Such changes could lead to frameshifts or base pair replacement leading to the introduction of stop codons, and may remove the activity of the encoded protein when the gene sequence is still present in the genome. In addition, these changes may produce pseudogenes. Since the changes are not random, the pseudogenes may be over-presented in certain functional classes such as pathogenicity island and cell-associated genes. For example, *S.typhi* CT18 and Ty2 contain inactivated genes which are involved in virulence and host range. For *S. typhimurium*, several genes that have been shown to be important for phenotypes in *S. typhimurium* appear to be inactive in *S.typhi* [19]. Therefore, further studies of *S.typhi* are likely to reveal rearrangements, insertions, translocations, and horizontal transfers corresponding to different tissue specificity and pathogenic effects for human and other organisms. Potentially the alteration of transcription and translation between related strains needs to be checked and confirmed by wet-bench genetic analysis. We think that although comparative genomics can provide very large amount of information on variations in each genome, it is still only an initial step in understanding the biology of an organism. Analysis of the complete genome sequence is only the start of the biological journey. The C++ and Matlab scripts for wavelet analysis and cumulative diagrams (Keto and purine excesses) are available on request from authors.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees and also Prof. C. Sensen for his comments on earlier versions of this paper. They would also like to thank Dr. Doug Phillips for his computer support.

REFERENCES

[1] R. D. Fleischmann, M. D. Adams, O. White, et al., "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd.," *Science*, vol. 269, no. 5223, pp. 496–512, 1995.

[2] E. V. Koonin and M. Y. Galperin, "Prokaryotic genomes: the emerging paradigm of genome-based microbiology," *Current Opinion in Genetics & Development*, vol. 7, no. 6, pp. 757–763, 1997.

[3] R. Himmelreich, H. Plagens, H. Hilbert, B. Reiner, and R. Herrmann, "Comparative analysis of the genomes of the bacteria Mycoplasma pneumoniae and Mycoplasma genitalium," *Nucleic Acids Research*, vol. 25, no. 4, pp. 701–712, 1997.

[4] M. McClelland, L. Florea, K. Sanderson, et al., "Comparison of the Escherichia coli K-12 genome with sampled genomes of a Klebsiella pneumoniae and three salmonella enterica serovars, Typhimurium, Typhi and Paratyphi," *Nucleic Acids Research*, vol. 28, no. 24, pp. 4974–4986, 2000.

[5] J. M. Freeman, T. N. Plasterer, T. F. Smith, and S. C. Mohr, "Patterns of genome organization in bacteria," *Science*, vol. 279, no. 5358, pp. 1827–1829, 1998.

[6] J. R. Lobry, "Asymmetric substitution patterns in the two DNA strands of bacteria," *Molecular Biology and Evolution*, vol. 13, no. 5, pp. 660–665, 1996.
[7] A. Grigoriev, "Analyzing genomes with cumulative skew diagrams," *Nucleic Acids Research*, vol. 26, no. 10, pp. 2286–2290, 1998.
[8] A. Grigoriev, "Strand-specific compositional asymmetries in double-stranded DNA viruses," *Virus Research*, vol. 60, no. 1, pp. 1–19, 1999.
[9] P. Lio, "Wavelets in bioinformatics and computational biology: state of art and perspectives," *Bioinformatics*, vol. 19, no. 1, pp. 2–9, 2003.
[10] A. Arneodo, B. Audit, E. Bacry, S. Manneville, J.-F. Muzy, and S. G. Roux, "Thermodynamics of fractal signals based on wavelet analysis: application to fully developed turbulence data and DNA sequences," *Physica A*, vol. 254, no. 1-2, pp. 24–45, 1998.
[11] J. Song, A. Ware, and S.-L. Liu, "Wavelet to predict bacterial ori and ter: a tendency towards a physical balance," *BMC Genomics*, vol. 4, no. 1, pp. 17, 2003.
[12] X.-Y. Zhang, Y.-T. Zhang, S. C. Agner, et al., "Signal processing techniques in genomic engineering," *Proceedings of the IEEE*, vol. 90, no. 12, pp. 1822–1833, 2002.
[13] W. Deng, S.-R. Liou, G. Plunkett III, et al., "Comparative genomics of Salmonella enterica serovar Typhi strains Ty2 and CT18," *Journal of Bacteriology*, vol. 185, no. 7, pp. 2330–2337, 2003.
[14] R. A. Alm, L. S. Ling, D. T. Moir, et al., "Genomic-sequence comparison of two unrelated isolates of the human gastric pathogen Helicobacter pylori," *Nature*, vol. 397, no. 6715, pp. 176–180, 1999.
[15] J.-F. Tomb, O. White, A. R. Kerlavage, et al., "The complete genome sequence of the gastric pathogen Helicobacter pylori," *Nature*, vol. 388, no. 6642, pp. 539–547, 1997.
[16] A. S. Wunenburger, A. Colin, J. Leng, A. Arneodo, and D. Roux, "Oscillating viscosity in a lyotropic lamellar phase under shear flow," *Phys. Rev. Lett.*, vol. 86, no. 7, pp. 1374–1377, 2001.
[17] S. G. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, London, UK, 1999.
[18] J. A. Abildskov, "Additions to the wavelet hypothesis of cardiac fibrillation," *Journal of Cardiovascular Electrophysiology*, vol. 5, no. 7, pp. 553–559, 1994.
[19] J. Parkhill, G. Dougan, K. D. James, et al., "Complete genome sequence of a multiple drug resistant Salmonella enterica serovar Typhi CT18," *Nature*, vol. 413, no. 6858, pp. 848–852, 2001.

Jiuzhou Song received his Ph.D. degree in statistical genetics from China Agricultural University in 1996. From 1996 till 1998, he held a postdoctoral fellowship in genetics at Hebrew University, and from 1998 till 2000, he was a Research Fellow in biochemistry and molecular biology at the Indiana University. Now he is a Research Associate in the Departments of Microbiology & Infectious Disease, and Biochemistry & Molecular Biology, Faculty of Medicine, University of Calgary. His main work is on bioinformatics and statistics, especially on high throughput gene expression data analysis, comparative genomics, biopathway and gene discovery, gene network, regulatory analysis, phylogenetic domain analysis, and computational biology.

Tony Ware received his Ph.D. degree in numerical analysis from Oxford University in 1991, having five years earlier obtained an honours degree in mathematics (First Class). From 1991 till 1993, he held a research fellowship in Oxford, and from 1993 till 1997, he was a Lecturer in applied mathematics at the University of Durham, UK. From 1997 till 1998, he received a research fellowship from the Department of Clinical Neurosciences at the University of Calgary. Since 2000, he has been an Assistant Professor in the Department of Mathematics and Statistics at the same university.

Shu-Lin Liu received his Ph.D. degree from Gifu University in 1990. He is an Adjunct Assistant Professor in the Department of Microbiology & Infectious Diseases, Faculty of Medicine, University of Calgary, Canada. His research focuses on bacterial evolution and speciation and is currently supported by grants from the Canadian Institutes of Health Research (CIHR) and Natural Science and Engineering Research Council of Canada.

M. Surette has been a Canada Research Chair in Microbial Gene Expression and an Alberta Heritage Foundation for Medical Research Senior Scholar since 2002. He is an Associate Professor in the Departments of Microbiology & Infectious Disease, and Biochemistry & Molecular Biology, Faculty of Medicine, University of Calgary, Canada. He has received Young Investigator Awards from Bio-Mega/Boehringer Ingelheim (Canada) in 1998–2001 and the 2000 Fisher Award from the Canadian Society of Microbiologists. His research focuses on population behaviors in bacteria and high throughput gene expression methods applied to studying bacterial virulence. His work is currently supported by grants from the Canadian Institutes of Health Research (CIHR), the Canadian Bacterial Disease Network, Genome Canada, the Human Frontiers Science Program, and Quorex Pharmaceuticals.

EURASIP Journal on Applied Signal Processing 2004:1, 13–28

Autoregressive Modeling and Feature Analysis of DNA Sequences

Niranjan Chakravarthy
Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-5706, USA
Email: niranjan.chakravarthy@asu.edu

A. Spanias
Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-5706, USA
Email: spanias@asu.edu

L. D. Iasemidis
Harrington Department of Bioengineering, Arizona State University, Tempe, AZ 85287-9709, USA
Email: leon.iasemidis@asu.edu

K. Tsakalis
Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-5706, USA
Email: tsakalis@asu.edu

Received 28 February 2003; Revised 15 September 2003

A parametric signal processing approach for DNA sequence analysis based on autoregressive (AR) modeling is presented. AR model residual errors and AR model parameters are used as features. The AR residual error analysis indicates a high specificity of coding DNA sequences, while AR feature-based analysis helps distinguish between coding and noncoding DNA sequences. An AR model-based string searching algorithm is also proposed. The effect of several types of numerical mapping rules in the proposed method is demonstrated.

Keywords and phrases: DNA, autoregressive modeling, feature analysis.

1. INTRODUCTION

The complete understanding of cell functionalities depends primarily on the various cell activities carried out by proteins. Information for the formation and activity of these proteins is coded in the deoxyribonucleic acid (DNA) sequences. For detection purposes, the vast amount of genomic data makes it necessary to define models for DNA segments such as the protein coding regions. Such models can also facilitate our understanding of the stored information and could provide a basis for the functional analysis of the DNA. Since the DNA is a discrete sequence, it can be interpreted as a discrete categorical or symbolic sequence and hence, digital signal processing (DSP) techniques could be used for DNA sequence analysis. The DNA sequence analysis problem can be considered as analogous to some forms of speech recognition problems. That is, coding and noncoding regions in DNA need to be identified from long nucleotide sequences, a process that bears some similarities to the problem of identifying phonemes from long sequences of speech signal samples. Currently proposed DSP techniques include the study of the spectral characteristics [1, 2, 3, 4] and the correlation structure [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] of DNA sequences. The measurement of spectra in most cases has been characterized by nonparametric Fourier transform techniques [1]. In some of the most common cases, the presence of a spectral peak [1] was used to characterize protein-coding regions in the DNA. On the other hand, correlations have been often characterized on the basis of the extent of power-law (long-range) behavior and the persistence of the power-law correlation sequence [6, 8]. Attempts have been also made to parameterize these correlations in terms of the scale of the power law [6].

In this paper, we propose the use of parametric spectral methods for the analysis of DNA sequences. Parametric spectral analysis techniques have been widely used to study time series of speech, seismic, and other types of signals. Specifically, we investigate the use of autoregressive (AR) spectral

estimation tools for DNA sequence analysis. AR models effectively capture spectral peaks and model the correlation in sequences [19]. After the model fit, the AR model parameters, and AR related signals such as the prediction residual, can be used as features of the DNA sequences. The studies that we carried on AR models include the following. First, we explored the use of linear prediction residuals to compare coding and noncoding regions as well as distinguish between different genes. Different numerical mapping rules for the representation of nucleotides were considered. Second, we used the AR parameters as DNA sequence features.

The paper is organized as follows. A few basic biological properties of the DNA are described in Section 2. An overview of DNA sequence analysis techniques based on correlation functions and DSP-based methods is presented in Section 3. The motivation for the use of parametric spectral analysis methods for DNA analysis and its various implementation aspects are presented in Section 4. Results from the application of AR model-based analysis to DNA sequences are presented in Section 5. A discussion of the results and possible extensions to these techniques are given in Section 6.

2. DNA STRUCTURE AND FUNCTION

DNA is the basic information storehouse in living cells. Various cell activities are carried out by proteins which are produced based on information stored in genes. DNA is a polymer formed from 4 basic subunits or nucleotides, namely, adenine (A), cytosine (C), thymine (T), and guanine (G). A single DNA strand is formed by the covalent bonds between the sugar phosphate groups of the nucleotides. Two DNA strands are then weakly bonded by hydrogen bonds between the nucleotides. Since the nucleotide A forms such a bond only with T, and G only with C, the two DNA strands are complementary to each other and each of them is used as a template during cell division to transfer information. Usually, two complementary DNA strands form a double helix. The synthesis of proteins is governed by certain regions in the DNA called protein *coding regions* or genes. The 64 possible nucleotide triplets ((nucleotide alphabet size)$^{\text{word length}} = 4^3$), called *codons*, are mapped into 20 amino acids that bond together to form proteins. Certain codons known as start and stop codons indicate the beginning and end of a gene. The DNA also consists of regions that store information for regulatory functions. In advanced organisms, the protein coding regions are not generally continuous and are separated into several smaller subregions called exons. The regions between the exons are known as introns. During the protein coding process, these introns are eliminated and the exons are spliced together. The splicing can be carried out in a number of different ways depending on the cell function. Splicing thus also determines the type of protein synthesis and hence genes can be used for the production of a variety of proteins. The central dogma (Figure 1) in cellular biology describes the information transfer from the DNA to the ribonucleic acid (RNA) and the production of proteins. The formation of proteins takes place in two stages, namely, transcription and translation. During transcription, the genes in the DNA sequence are used as templates to form the pre-messenger RNA (pre-mRNA). The pre-mRNA is a polymer formed from 4 basic subunits, namely, A, C, G, and uracil (U). Next, the exons in the pre-mRNA are spliced together to form a polymer of only coding regions known as the mRNA. The mRNA along with the transfer RNA (tRNA) controls protein formation. The complete process is controlled and catalyzed by a number of enzymes. Almost all cells in a living system have the same DNA structure and information content. The gene expression depends on the cell requirements. Microarray technology basically captures the amount of expression of various genes. The structure and organization of the DNA and various cell functions are explained in [20].

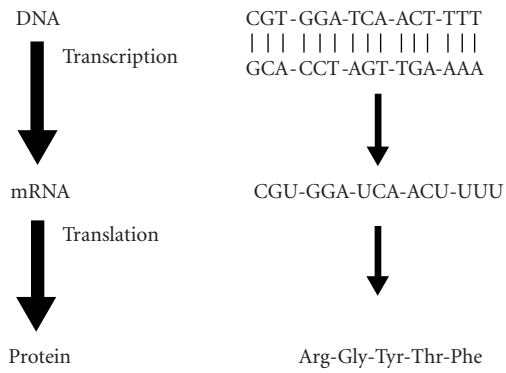


Figure 1: Central dogma; the information transfer from DNA to proteins.

One of the relevant problems in bioinformatics is to accurately identify the protein coding regions and thus predict the protein that will be generated using the information in these segments. In addition, some effort is expended in understanding the role of noncoding regions. It is therefore of central interest to analyze and characterize various DNA regions such as coding and noncoding sequences.

3. REVIEW OF METHODS FOR DNA SEQUENCE ANALYSIS

A primary objective of DNA sequence analysis is to automatically interpret DNA sequences and provide the location and function of protein coding regions. Methods to locate genes, and various coding measures are described in [21]. The gene identification problem is challenging especially in eukaryotic DNA sequences in which the coding regions are separated into several exons. An overview of standard techniques for gene identification is provided in [22]. Computational techniques for gene identification are classified into template methods and lookup methods. Template methods attempt to model prototype objects or sequences and identify genes based on these models. On the other hand, lookup methods use exactly known gene sequences and search for similar segments in a database. Computational techniques, to accomplish the above, include identification measures like Fourier spectra and sequence similarity measures. An overview of the

standard coding measures and their accuracy in identifying genes is also given in [22]. A discussion on the regulation of gene expression, techniques to integrate various gene models, for example, hidden Markov models (HMM), and methods for efficient computation are presented in [22] as well.

3.1. Correlations in DNA sequences

Correlation functions have been widely used to study the statistical properties of DNA sequences. The autocorrelation of a stationary and ergodic numerical sequence x at lag m is defined as

$$\begin{aligned} r_{xx}(m) &= E[x(n+m)x(n)] \\ &= \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n+m)x(n), \end{aligned} \tag{1}$$

where $E[\cdot]$ is the statistical expectation operator and N is the length of the window over which the averaging is performed. A typical statistically well-behaved estimator for the autocorrelation is

$$\hat{r}_b(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n+|m|)x(n). \tag{2}$$

The power spectrum of a signal is the Fourier transform of its correlation [19]. To use (2) in DNA analysis, one has to assign numerical values to the nucleotides A, T, C, and G. One of the early analyses of the correlation structure in the DNA was done in [6]. Binary indicator sequences are used therein to calculate correlations in the DNA sequence. The power spectra of the sequences are shown to have a power-law behavior. The spectra are reported to change according to the evolutionary categories of the DNA sequences analyzed. Similar analysis is also presented in [11], wherein a simple model, called expansion-modification model, is considered to exhibit correlations similar to those present in the DNA. Results are therein presented based on three correlation measures, that is, the mutual information function, the power spectrum to calculate the correlations, and a cumulative approach (similar to a DNA walk). Various issues of the DNA correlation structure and its interpretation are also discussed.

The calculation and relation between correlation functions and mutual information of symbol sequences are explained in [5]. Correlation functions and mutual information function differ in quantifying statistical dependencies. While correlations measure only the linear dependencies in sequences, the mutual information function detects other statistical dependencies (e.g., nonlinear) in the signal as well. The correlation measurements depend on the assignment of numbers to the symbols in the sequence, whereas the mutual information is independent of such coordinate transformations. The binary mapping rules used in [7] carry certain biological interpretations and are used in the calculation of the autocorrelation and the other related statistical dependencies. A study on the statistical correlations in the DNA sequence is presented in [8], in which possible errors in estimating correlations from short DNA sequences is also described. The direct measure of correlations from long sequences is advocated to be better than measures obtained through detrended fluctuation analysis (DFA) [10], indirect autocorrelation computation from the power spectra, and correlation estimates from the mutual information function [11]. The DFA technique removes heterogeneities in the DNA sequence, but since it has been reported that important details of the correlation structure in the DNA may be due to these heterogeneities [23], the use of the DFA technique is questioned. The autocorrelation function is considered to be useful in measuring the compositional heterogeneity. A series of studies on the use of correlation in DNA analysis is also given in [9, 14, 15, 16, 17, 18]. Other methods for DNA analysis include DNA walk [24] and Markov chains of various orders.

Observed correlation properties have also been interpreted in terms of the underlying biology [11, 12, 13, 18]. One of the important characteristics of protein coding segments in DNA sequences is the presence of persistent correlations with a pronounced period of three. It is shown in [12] that these correlations arise due to the nonuniform usage of codons in the coding regions. This nonuniformity is considered to exist due to a number of factors including the many-to-one mapping of codons to amino acids, the use of certain amino acids for protein formation, the preferential coding of codons into amino acids, and the correlations between the G + C contents in the third codon positions with G + C contents in the surrounding DNA. These factors may cause the concentrations of nucleotides in the three codon positions to be different. Such a positional asymmetry is believed to be the cause of the pronounced period-three pattern in the coding segment correlations and mutual information. The pronounced periodicity mentioned in [12] has also been used to differentiate coding and noncoding DNA segments [25]. Covariance matrix decay is used for analysis of correlation functions in [13]. The observations of long-range correlations and the various periodicities in the observed correlations are related to biological facts in genomes.

The characterization of coding and noncoding regions based on the mutual information function is described in [25]. That paper basically explores the existence of phylogenetic origin-free statistical features in coding and noncoding regions. The mutual information function decays to zero for noncoding DNA, whereas it oscillates for coding DNA with a period of three. Gene identification based on the mutual information function is reported to perform better than traditional techniques which require training on datasets [26]. A number of other information theory measures have also been used for coding segment characterization [5, 18, 23, 27, 28, 29, 30, 31]. A measure for sequence complexity is presented in [23]. The sequence compositional complexity is based on an entropic segmentation method to divide a sequence into homogenous segments. The complexity measure is compared for coding and noncoding segments and is related to the correlation structure. An entropic segmentation method is also used in finding borders between coding and noncoding regions [27]. A 12-letter alphabet or mapping rule is used, which takes into account the

differential base composition at each codon position. This is used to find different compositional domains for coding and noncoding regions. General statistical properties of coding regions are used in the segmentation, and this method is reported to be highly accurate in identifying borders. Another information theory tool which has been reported to be useful in the analysis of DNA sequences is given in [28]. This is the Jensen-Shannon divergence which quantifies the difference between different statistical distributions. A description of statistical properties of the divergence measure is followed by the application to the analysis of DNA sequences. The segmentation method based on the divergence measure is reported to segment a nonstationary sequence into stationary subsequences, and is also applied to DNA. Finally, a good overview on information theory and applications to molecular biology can be found in [32].

3.2. DSP techniques for DNA sequence analysis

The string of nucleotides in the DNA sequence is a categorical or symbolic sequence. Each of the nucleotides is assigned a numerical value, in order to apply DSP methods. Examples of such numerical assignment techniques are the binary indicator sequences [6] or the assignment of the integers 1, 2, 3, and 4 to A, C, G, and T, respectively [33]. The numerical sequences thus obtained are analyzed using DSP methods. Tiwari et al. [1] identify coding regions in DNA sequences by computing the Fourier spectra of a moving window across the sequence. The value of the spectrum at $f = 1/3$, is used to clarify the DNA regions as either coding or noncoding. The relative strength of the periodicity is used as the coding measure (ratio of the spectral value at $f = 1/3$ to the average spectrum). The effectiveness of the GeneScan method in identifying coding regions is also discussed. The method is robust to sequencing errors resulting from frameshift errors; the computations are simple and training is not required, which is an additional advantage. Anastassiou [2] extends on the ideas from [1, 3] and provides a method to differentiate coding and noncoding regions based on weighted spectra. Two numerical assignment schemes, namely, binary and complex number assignments are used for analysis in [2]. A procedure to compute the protein sequence from the coding regions, based on the principles of finite impulse response filters and quantization, is also described. Methods to calculate DNA spectrograms, and the use of power spectra to identify coding regions, are given. The paper also describes the method for the identification of reading frames and summarizes the uses of DSP-based techniques in DNA sequence analysis. Analysis of chromosome genomic signals has also been carried out using a complex numerical representation of nucleotides [34]. Therein, a model of the structure of the chromosome has been presented through techniques such as phase analysis, two- and three-dimensional sequence path analysis, and statistical analysis. The signal processing of symbolic sequences has also been addressed in [35, 36]. In [35], binary indicator sequences are used for DNA sequence analysis. For any mapping rule, a symbolic sequence is mapped to a numerical sequence by assigning a weight to each symbol. This mapping can be represented as a matrix multiplication. The subsequent linear transformation of the numerical sequence can also be represented by a matrix multiplication operation. Since linear transformations are performed, the weights can be optimized to obtain a required property in the transformed signal. These operations are explained in the case of discrete Fourier transforms (DFTs). The computation of linear transforms for symbolic signals is also explained in [36]. Spectral and wavelet analyses of symbolic sequences are explained and applied to DNA sequences, and results are presented for "pseudo DNA" sequences and *E. Coli* DNA.

Concepts from digital IIR filtering were used in [4] to detect coding regions. This paper uses antinotch IIR filters to identify these regions. This is achieved by designing a filter which has a sharp frequency response peak at $2\pi/3$. On passing the nucleotide sequence through this filter, if the sequence is from a coding region, the output will have a pronounced frequency peak at $2\pi/3$. The authors explain various tradeoffs in the design of the IIR filter and efficient design procedures. They conclude with examples where the output of the antinotch filter has a more discernible spectral peak at $2\pi/3$ when coding sequences are analyzed.

Two DSP-based approaches to genome sequences analysis are explained in [24]. The methods are the three-dimensional DNA walks and Gauss wavelet-based analysis, and Huffman-based encoding technique. The three-dimensional DNA walk is used as a tool to visualize changes in nucleotide composition, base pair patterns, and evolution along the DNA sequence. The proposed DNA walk model is reported to provide similar results as those obtained from a purine-pyrimidine walk, in terms of long-range correlations. Gauss wavelet analysis is then used to analyze the fractal structure of the three-dimensional DNA walk. With the use of Huffman coding, the transformation of the DNA sequence into an encoded domain can help visualize the sequences from a new perspective.

The spectral analysis of a categorical time series is explained in [37, 38]. In [37], the statistical theory for analyzing a categorical time series in the frequency domain is discussed, and the methodology that is developed is applied to DNA sequences. A discussion on the application of the spectral envelope methodology to a number of sequences, including the DNA, is given in [38]. Various spectral peaks in the sequence can be observed in the spectral envelope that is obtained through this technique. Techniques based on time-frequency and wavelet analysis have also been used to analyze DNA and protein sequences [18, 39, 40, 41].

3.3. Numerical mapping of nucleotides

Numerical mapping can be broadly classified into two types, namely, fixed mapping as in [1, 2, 4, 5, 6, 7, 8, 13, 16, 17, 24, 33] and a mapping based on some optimality criterion as in [36, 37]. Fixed mappings include binary [8], integer [33], and complex representations [2]. In this work, we use a real-number mapping rule based on the complement property of the complex mapping in [2]. The real-number representation is A $= -1.5$; T $= 1.5$; C $= 0.5$; and G $= -0.5$.

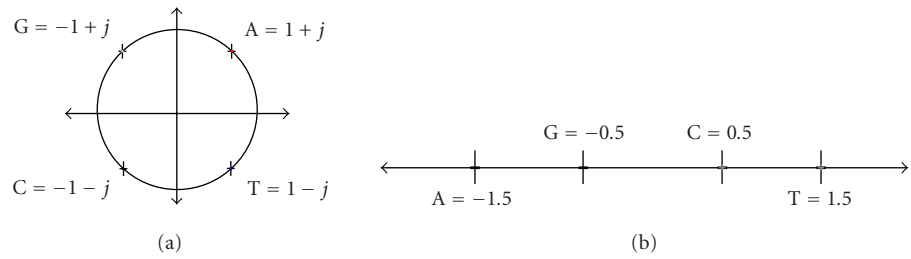


Figure 2: A constellation diagram for (a) complex-number representation and (b) real-number representations.

The complement of a sequence of nucleotides can be obtained by changing the sign of the equivalent number sequence and reversing the sequence. For example, CTGAA: 0.5; 1.5; −0.5; −1.5; −1.5 → Change Sign and Reverse Sequence → 1.5; 1.5; 0.5; −1.5; −0.5: TTCAG. In the computation of correlations, real representations are preferred over complex representations. Furthermore, it is interesting to note that the complex, real, and integer representations can also be viewed as constellation diagrams, which are widely used in digital communications. Figure 2 shows the constellation diagram for the complex and real representations. The complex constellation is similar to that of the quadrature phase shift keying (QPSK) scheme, and the real representation is similar to the pulse amplitude modulation (PAM) scheme. The constellation diagram helps visualize the DNA sequence in the context of digital communications, where a symbol mapping is followed by transmission of information. Analysis of DNA sequences using digital communications techniques could reveal certain aspects of the DNA like error-correcting capability. An information theory perspective of information transmission in the DNA, namely, the central dogma, is explained in [32].

4. AR MODEL-BASED DNA SEQUENCE ANALYSIS

The aforementioned DNA sequence analysis techniques can be divided into two main categories. In the first category, correlations within coding and noncoding sequences are characterized and used thereafter. In the second category, the Fourier transform of sequences is used to observe spectral characteristics that could distinguish between coding and noncoding DNA regions. The typical spectral signature found in a coding region is a spectral peak [1], and AR spectral estimators are effective in modeling spectral peaks of short sequences [19]. AR spectral parameters can also reflect the underlying difference in the correlation structure between coding and noncoding regions. Since correlations have been related to biological properties of the DNA, AR models could also be used as models of biological functions. Hence, it is a logical extension to use AR spectral estimators to analyze DNA sequences.

4.1. AR modeling

The AR modeling of DNA sequences can be performed using linear prediction techniques. In the linear prediction analysis, a sample in a numerical sequence is approximated by a linear combination of either preceding or future sequence values [42]. The forward linear prediction operation is given by

$$e(n) = x(n) - a_1 x(n-1) - a_2 x(n-2) - \cdots - a_p x(n-p), \tag{3}$$

where x is the numerical sequence, n is the current sample index, $a_1, a_2, \ldots, a_p$ are the linear prediction parameters, and $e(n)$ is the linear prediction error. Equation (3) represents forward linear prediction since the current sample is predicted by a linear combination of previous samples. Similarly, in backward linear prediction, a sample is predicted as a linear combination of future samples. The linear prediction coefficients are calculated by minimizing the mean squared error. The linear prediction polynomial is given by

$$A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i}. \tag{4}$$

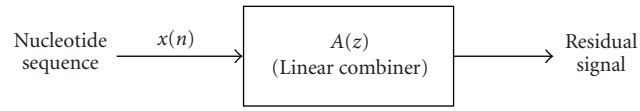


Figure 3: AR process and linear prediction; $A(z)$ is the filter polynomial.

Figure 3 depicts the DNA linear prediction in the context of AR processes.

The output of the linear combiner is known as the residual signal. In speech processing, linear prediction has been used for efficient modeling with a considerable level of success [43]. The AR Yule-Walker and Burg algorithms are widely used to compute the AR model parameters. The involved autocorrelation matrix values are typically calculated using the biased estimate in (2). Issues related to the AR modeling of DNA sequences are discussed in Section 4.2.

4.2. Proposed AR model-based DNA sequence analysis

The AR modeling of a DNA sequence is done by first mapping the sequence into the numerical domain and then calculating the AR parameters of the resulting numerical sequence. Since the numerical mapping of the DNA affects

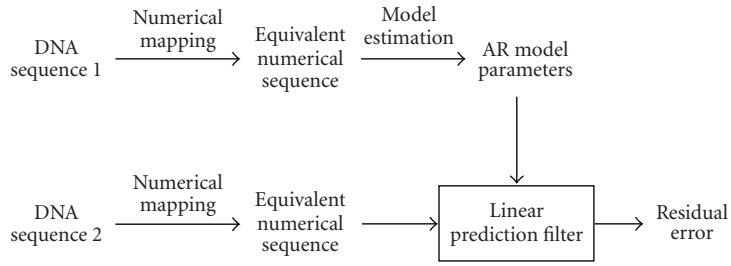


Figure 4: Block diagram of AR model-based residual signal analysis of DNA segments.

the correlation function [5], the AR parameters, which are derived from the correlation values, also depend on the numerical assignment. In this paper, the real, integer, and binary mapping rules [8] have been used for analysis. Another important issue pertains to the application of AR modeling to DNA sequences. As mentioned in Section 4.1, the calculation of AR parameters from the linear prediction model involves minimizing the error between the current signal sample and a linear combination of past samples. This definition pertains to causal AR modeling. In the case of DNA sequences, there appears to be no constraint to consider only a causal AR model, since the nucleotides in a spatial series need not be constrained to depend on the ones positioned before them only. However, the protein coding information is stored in nucleotide triplets and certain codons signal the start and stop of these gene regions. The start/stop codons and the transcription of the nucleotide triplets implicitly confer directionality to the nucleotide sequences in the genes. Hence, a causal AR model appears to be more appropriate for modeling gene sequences. The fact that the polymerase enzyme which is responsible for reading the information from the genes physically reads this DNA information from the start to the stop codons augurs our assumption. However, it needs to be noted that no such directionality apparently exists in noncoding regions and it would thus be of considerable interest to analyze both coding and noncoding DNA regions with causal versus noncausal models, respectively.

AR models of DNA sequences were used to perform two basic kinds of analyses. In the first analysis, the residual error variance of DNA sequences was used as a measure to indicate the "goodness" of the AR fit. In other words, AR models of various DNA segments were compared based on their AR residual signal. That is, suppose that signals $s_1(n)$ and $s_2(n)$ are modeled using respective AR models. When $s_1(n)$ is input to the linear predictor defined by the parameters of the AR model of $s_2(n)$, the residual signal error would be lower if $s_1(n)$ and $s_2(n)$ are described by similar AR models than if described by different AR models. The residual signal can thus be used as a measure of similarity between two signals (e.g., two DNA regions). Furthermore, it is evident that the residual error (a one-dimensional measure) alone is not sufficient to parameterize multidimensional signals, that is, different signals may yield similar residual error values. Thus, the inadequacy of the residual error was one of the motivations to use AR model parameters as sequence features. For example, if the parameters $a_1, a_2, \ldots, a_p$ are obtained by AR analysis of a gene segment, the vector $[1, a_1, a_2, \ldots, a_p]^{\mathrm{T}}$ is used as the segment feature. This is similar to the analysis of speech signals, where the AR model parameters or their derivatives, such as cepstral parameters, are used as feature vectors. Furthermore, by representing DNA sequences of different lengths with AR models of equal order, their comparison becomes possible by many simple measures such as Euclidean distance and vector correlations. Subsequently, AR features of coding and noncoding DNA sequences were analyzed using techniques such as feature space distribution analysis. Finally, we did not use the AR spectrum to distinguish between coding and noncoding features. This is due to the fact that working with high-order AR models, spurious spectral peaks were observed.

4.3. Analyzed DNA sequences

The analyses presented herein were performed on the *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, and *Streptococcus agalactiae* genomes. The *S. cerevisiae* genome has 16 chromosomes and its complete length is approximately 12 million bp. *C. elegans* and *C. cerevisiae* are eukaryotes, while *S. agalactiae* is a prokaryotic organism.

Prokaryotes are single-celled organisms while eukaryotes can be single- or multicelled. Major differences between prokaryotic and eukaryotic genomes are that the genome size of prokaryotes is typically less than that of eukaryotes, and that prokaryotic DNA has a higher percentage of genetic information content in contiguous gene segments than eukaryotic DNA. Furthermore, the number of repetitive sequences in eukaryote DNA sequences is larger than the number of repeats in prokaryote DNA. The above-mentioned genomes can be obtained from the National Center for Biotechnology Information (NCBI) public database.

5. RESULTS

5.1. Residual error analysis

We will first discuss the AR residual error-based DNA analysis. Results only from the analysis of *S. cerevisiae* chromosome 4 DNA sequence are presented herein. The binary SW mapping rule [8] and the real-number mapping rule were used. The analysis' block diagram is shown in Figure 4. AR models of coding and noncoding DNA regions were compared based on their AR residual errors as follows.

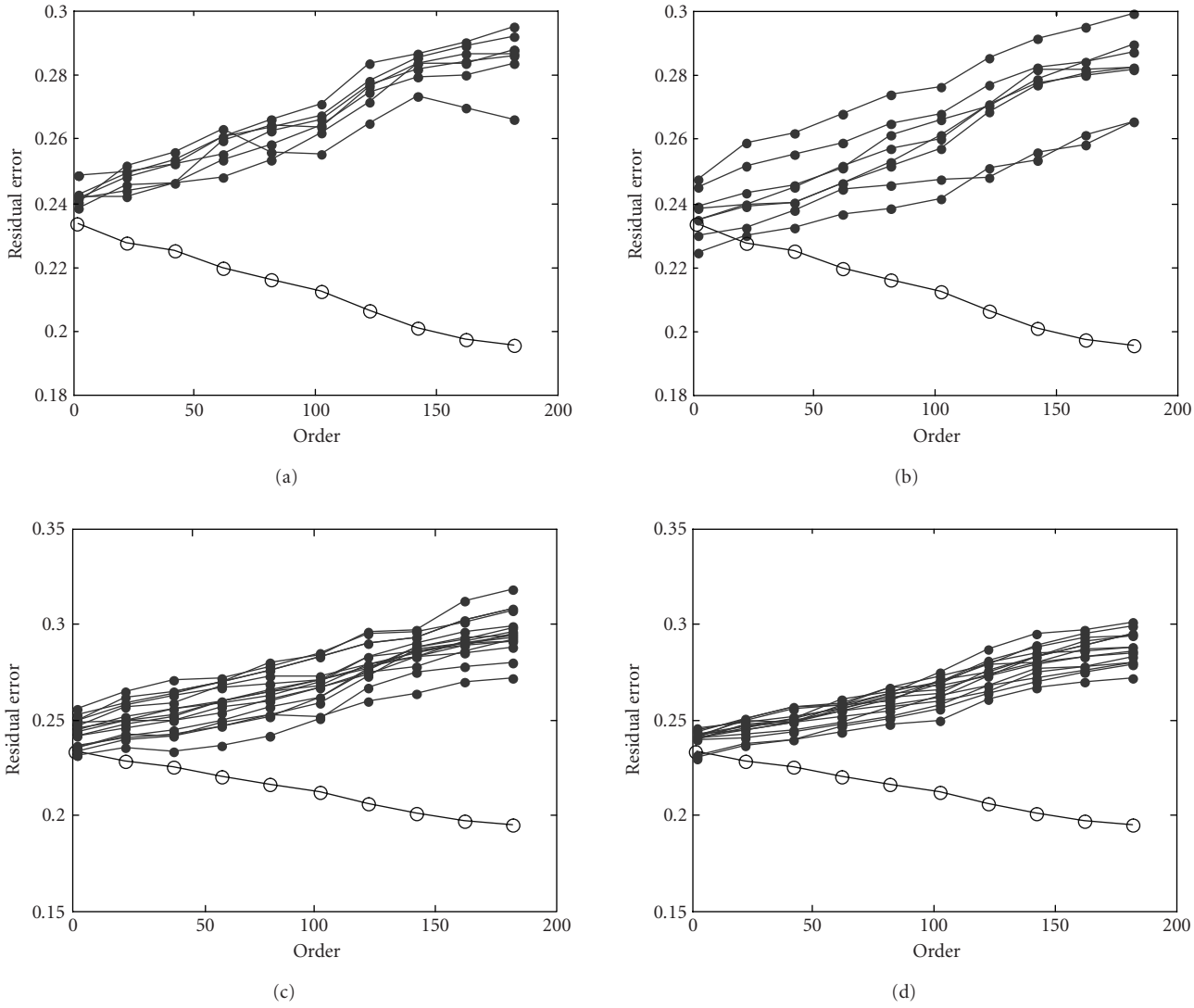


Figure 5: AR model of gene 1 of *S. cerevisiae* is used to perform residual signal analysis on its other genes using binary mapping. Residual signal variance versus AR model for gene 1 (-o-) and other genes (-•-) from chromosome 4, (a) error in gene 1 and genes 3–9; (b) error in gene 1 and genes 11–18; (c) error in gene 1 and genes 20–35; and (d) error in gene 1 and genes 36–50. Genes of length less than 150 bp were not considered since they cannot be modeled using high-order AR models.

First, the AR models were computed for each gene. Then, these AR model parameters were used to perform linear prediction and obtain the residual signal variances when applied to other genes. Genes of shorter length for which higher-order AR models could not be computed were not considered. The residual signal variances from 47 genes obtained with the AR model of gene 1 are shown in Figure 5. It can be noted that with increasing AR model order, the residual signal variance in gene 1 decreases. This is in conformance with the well-known fact from statistical signal processing that when a signal is modeled using AR models of increasing order, the residual signal error for that signal decreases monotonically [19]. On the other hand, it is interesting to note that for the other gene sequences, the residual error variance increases with increasing AR model order (see Figure 5). A similar result was observed when the real mapping rule was used (see Figure 6). This observation implies that with increasing model order, the similarity between the AR models of different genes decreases due to the increased specificity of the AR models to genes. The specificity could be due to the absence of redundancy between the analyzed genes and emphasizes the idea that, since different genes typically code for different amino acid sequences, they may not contain a lot of similar or redundant information.

Next, noncoding segments were compared with coding segments. Gene 1 in chromosome 4 of *S. cerevisiae* was modeled using an AR model, and the model parameters were used to compute the residual error variances of 50 noncoding

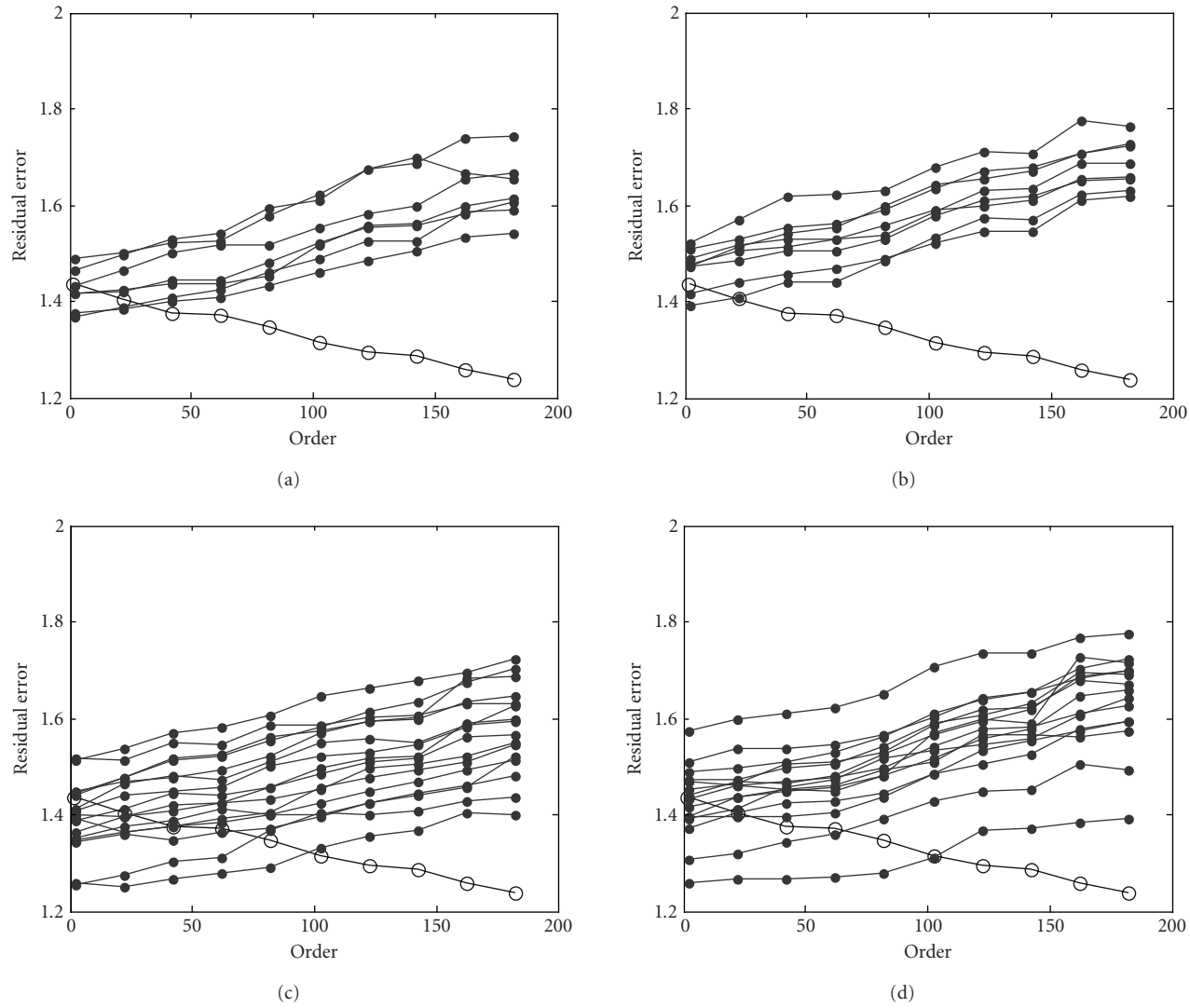


Figure 6: AR model of gene 1 of of *S. cerevisiae* is used to perform residual signal analysis on its other genes using real-number mapping. Residual signal variance versus AR model for gene 1 (-o-) and other genes (-•-) from chromosome 4, (a) error in gene 1 and genes 3–9; (b) error in gene 1 and genes 11–18; (c) error in gene 1 and genes 20–35; and (d) error in gene 1 and genes 36–50.

segments. Similarly, gene 17 was modeled using an AR model and the model parameters were used to compute the residual error variances of 50 noncoding segments. The residual error variances of 50 noncoding segments when the AR model from gene 1 and gene 17 was applied are depicted in Figures 7 and 8, respectively. It can be observed that the residual signal variance values for a few noncoding sequences are smaller than the ones for gene 1, for the full range of model orders. This implies the existence of similarities between coding and noncoding segments. Similar observations were also obtained when real mapping was applied.

It is evident from the above observations that the classification of an analyzed sequence to either a coding or noncoding region based on the residual signal alone is difficult as different regions may have similar residual errors for a range of AR model orders. The above results also show that when AR models are used to parameterize DNA segments based on the residual error, higher-order models may be required to model the characteristics and capture their differences.

5.2. AR feature-based analysis

One of the important problems in DNA sequence analysis is identifying regions with similar nucleotide compositions. This is then typically applied in studies such as identifying conserved regions across different organisms. A number of algorithms, such as BLAST, have been developed to perform string searches and template matching. These string searching tools are typically based on dynamic programming concepts, wherein the actual template or query string is compared with segments of a long DNA sequence. In this paper,

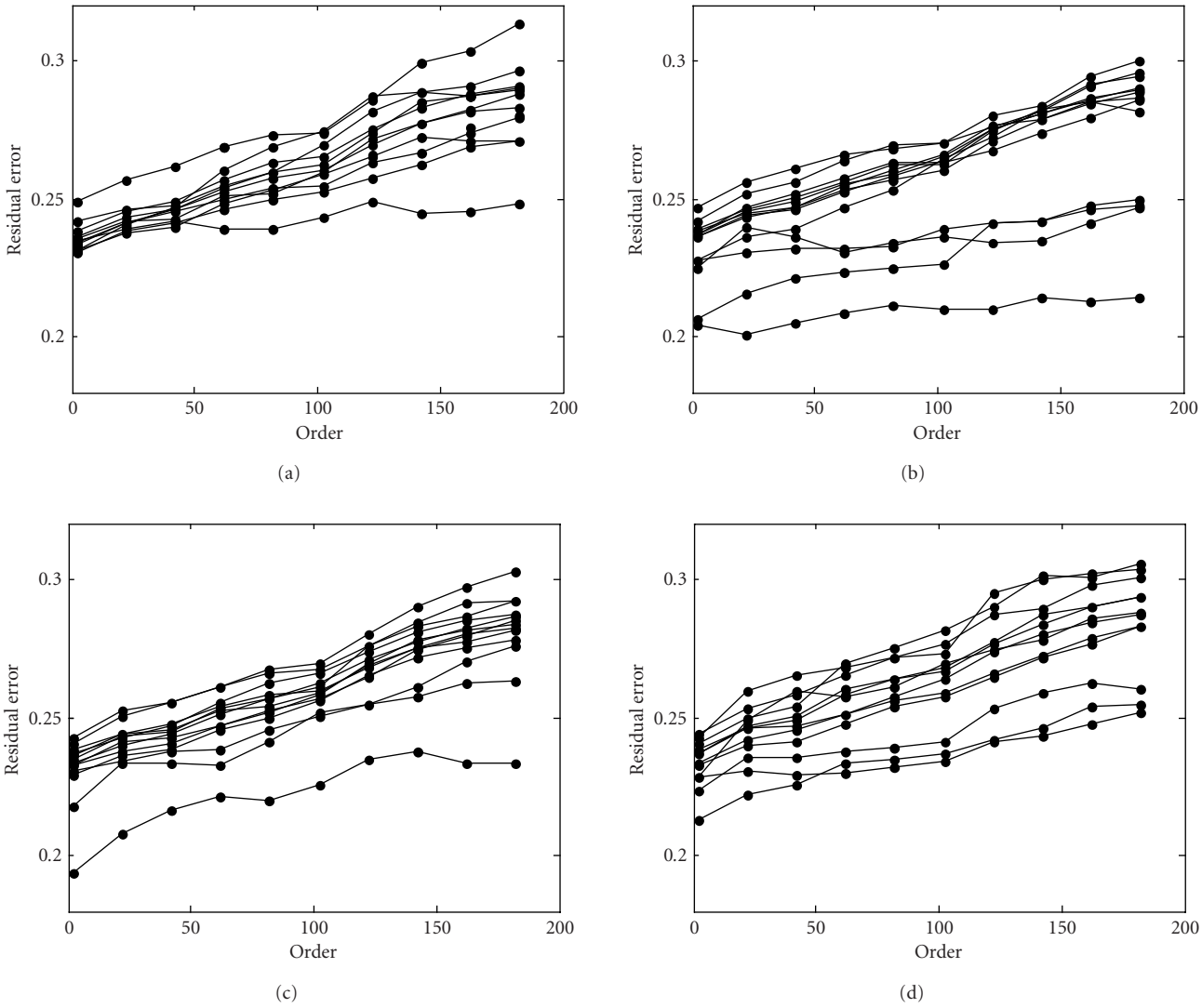


Figure 7: AR model of gene 1 is used for linear prediction on 50 noncoding segments using binary mapping. (a) Error in noncoding segments 1–12; (b) error in noncoding segments 13–25; (c) error in noncoding segments 26–38; and (d) error in noncoding segments 39–50.

the AR model parameters of the template nucleotide sequence are used as features to identify similar segments in a long DNA sequence. AR models capture the global spectral characteristics of the modeled sequences. Thus, the identification is based on similar spectral characteristics (AR) rather than one-to-one nucleotide matching (dynamic programming techniques).

The analysis was performed on a segment of the *S. cerevisiae* genome using binary, real-number, and integer mapping. The template matching procedure was performed as follows. First, a segment of nucleotides of length L was chosen as the template. The AR model of this template was estimated for various orders, and the model parameters were used as template features. Second, the AR features were calculated over the whole DNA sequence from overlapping moving windows of the same length L as the template. Third, the feature vectors obtained from each moving window were compared with the template feature vector by computing the Euclidean distance between them.

It was observed that using the real mapping, similar segments to either the template, its reversed sequence, its complementary sequence, or its reversed complementary sequence are detected. One such example is presented in Table 1, wherein the template and its complement were identified. Using integer mapping, the DNA locations where similar features were found are cited in Table 2. In this case, the features of the template sequence alone was detected. Using binary SW mapping, although the actual template occurred only once in the complete sequence, other segments also yielded the same features (see Table 3). Here the template and the matched sequences differ in the actual nucleotide but on a closer look, they have a similar sequence of strong and weak

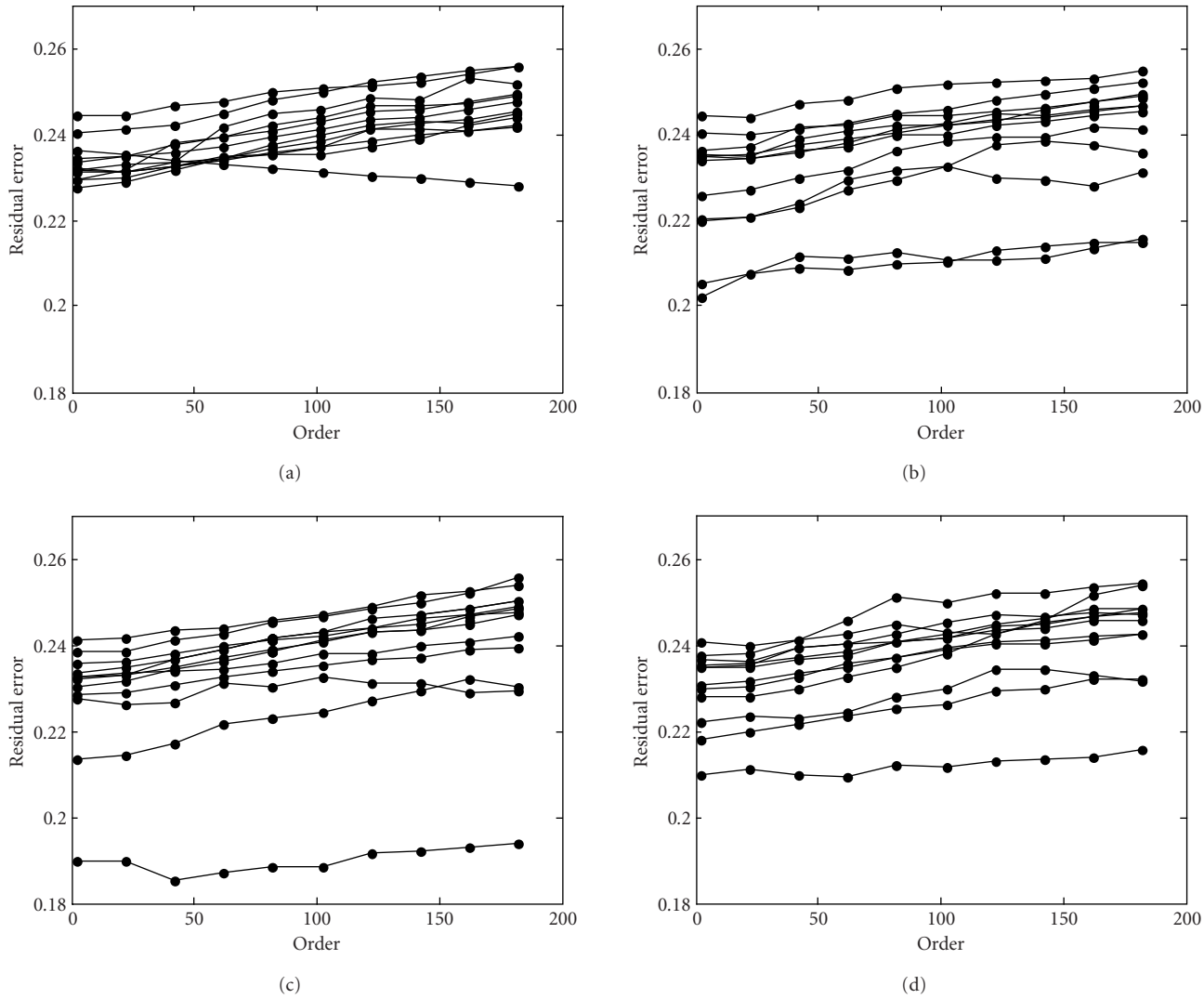


Figure 8: AR model of gene 17 is used for linear prediction of 50 noncoding segments using binary mapping. (a) Error in noncoding sequences 1–12; (b) error in noncoding sequences 13–25; (c) error in noncoding sequences 26–38; and (d) error in noncoding sequences 39–50.

hydrogen bonds. Analysis with the binary RY mapping rule [8] yielded similar results, that is, segments with a similar sequence of purines and pyrimidines as the one in the template.

In the aforementioned analysis, the mapping rule used played an important role in identifying matches. The real- and integer-number mapping rules yielded different string matches. This is due to the inherent complementary property of the real mapping rule and the noncomplementary property of the integer mapping rule. The difference is further elucidated through the following exercise. Say, for example, the occurrences of the template 5′-TACGTGC-3′ need to be found in a long DNA string. The corresponding numerical sequence obtained through real mapping would be 5′-1.5, −1.5, 0.5, −0.5, 1.5, −0.5, 0.5-3′. The following numerical sequences will have the same AR parameters as the above template:

(i) 5′- −1.5, 1.5, −0.5, 0.5, −1.5, 0.5, −0.5-3′ = 5′-ATGCACG-3′: (reversed complement of the template);

(ii) 5′-0.5, −0.5, 1.5, −0.5, 0.5, −1.5, 1.5-3′ = 5′-CGTGCAT-3′: (reversed template);

(iii) 5′- −0.5, 0.5, −1.5, 0.5, −0.5, 1.5, −1.5-3′ = 5′-GCACGTA-3′: (complement of the template).

This is due to the fact that (a) the sign-reversed numerical sequence and the actual numerical sequence have the same linear dependence and hence the same AR parameters, and (b) minimizing the forward or the backward linear prediction error would theoretically yield the same AR model. This is observed with the Burg algorithm AR estimation, wherein

Table 1: Detection of repeats of DNA segments via AR modeling. Real mapping rule and second-order AR model features are used; the template is 8 bp long. There are 5 repeats in the whole sequence. Identification of complementary and reversed sequences is obtained as well.

Position with the same features	DNA segment
210–217 (template)	CTCACATT
5174–5181	CTCACATT
12572–12579	CTCACATT
19278–19285	AATGTGAG
29624–29631	CTCACATT
36387–36394	AATGTGAG
55805–55812	AATGTGAG
63106–63113	CTCACATT

Table 2: Detection of repeats of DNA segments via AR modeling. Integer mapping rule and second-order AR model features are used; the template is 8 bp long. There are 5 repeats in the whole sequence. The template is exactly identified.

Position with the same features	DNA segment
210–217 (template)	CTCACATT
5174–5181	CTCACATT
12572–12579	CTCACATT
29624–29631	CTCACATT
63106–63113	CTCACATT

Table 3: Detection of repeats of DNA segments via AR modeling. Binary SW mapping rule and fourth-order AR model features are used; the template is 14 bp long and it has one occurrence in the whole sequence. Identification of DNA with similar sequences of strong and weak hydrogen bonds is obtained. Nucleotides C and G (mapped to one), A and T (mapped to zero) are highlighted differently.

Position with the same features	DNA segment
210–221 (template)	C T C A C ATTA CCC TA
7424–7435	C T C T G AAAT GCC AT
9283–9294	G A C T G ATAA GGG TT
80726–80737	C A G T G ATAT CGG TA

both the forward and backward linear prediction errors are minimized together. In the case of the integer mapping rule ($A = 1, C = 2, G = 3, T = 4$), the corresponding numerical sequence of the template is $5'$-$4, 1, 2, 3, 4, 3, 2$-$3'$. The reversed sequence, namely, $2, 3, 4, 3, 2, 1, 4$, has the same AR model parameters as the template (by minimizing the forward and reverse prediction errors). On the other hand, the sequence corresponding to the complement of the template may not have the same AR model. Hence, using the integer mapping rule, the exact template and its reversed sequence are matched.

The features of the nucleotide segments are also affected by the use of the binary mapping rule. This is explained through the following example. The sequence $5'$-TGACAAGC-$3'$ is mapped to $5'$-$0, 1, 0, 1, 0, 0, 1, 1$-$3'$ using the binary SW mapping rule. The above numerical sequence also corresponds to $5'$-ACACATGG-$3'$, and a number of other nucleotide combinations. The AR model parameters of all these combinations are the same, and hence, it is possible to identify sequences with certain similar chemical properties like similar sequences of strong and weak hydrogen bonds.

The above observations are of great interest because they show that identification of regions with similar biological/chemical properties may be possible using AR feature-based template matching under different mapping rules. For example, the ability to identify a template and its complement can help in identifying genes in complementary strands as well, which may not be possible in a single "run" using traditional string searching tools. The AR model string search method can be used as an analytical tool to reveal additional information about the interrelations between different DNA sequences. The knowledge acquired by this analysis could be used in knowledge or rule-based methods. Two DNA signals with similar AR spectra are more related in a global manner than in a one-to-one nucleotide basis. In this sense, the above method can provide clues about similarities between apparently nonidentical DNA sequences that could then be used in the identification of the underlying biochemical mechanisms of such similarities. The results of AR model-based analysis are related to fast Fourier transform (FFT)-based methods. The pros and cons have to do with the well-known advantages and disadvantages of using parametric versus nonparametric signal processing methods (e.g., ability to analyze short versus long segments, computational speed, etc).

The above algorithm was also applied to gene searches in a long string of DNA. It was observed that the distance between the feature vectors is zero at the exact location of the gene even with an AR model of an order as low as 2. The distance between the gene sequence AR feature vector and the moving window AR feature vector is plotted for various feature dimensions (AR model orders) in Figure 9. It was also observed that the average distance between the gene feature vector and features of the moving windows increased with AR model order. It can be typically expected that the average distance between vectors tends to increase with increasing dimension. Nevertheless, in conjunction with our previous observations from the residual signal-based analysis, it appears that the increasing average distance of the gene features with the AR model orders may mainly be due to the greater specificity of the AR modeling to the presence of genes. To further investigate the above observations, a study of the distributions of coding and noncoding AR features was undertaken.

The complete *S. cerevisiae* genome with all coding and noncoding sequences was considered. We mapped the DNA segments into the numerical domain using the binary SW mapping rule. Then, the AR model parameters of all segments were calculated and used as the DNA segment features.

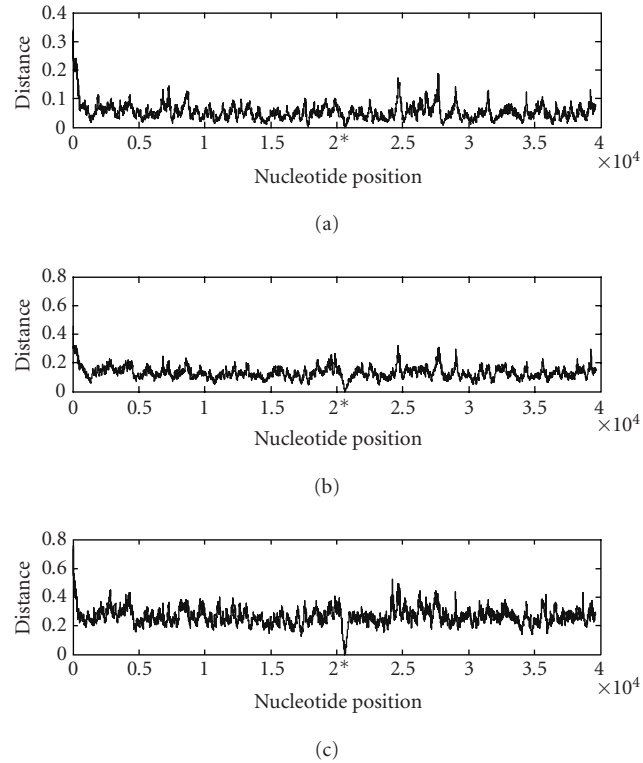


Figure 9: The distance between the feature vector of a gene sequence (position denoted by $*$) and the corresponding features within a moving window segments over the analyzed DNA sequence from *S. cerevisiae* for AR model orders (a) 10, (b) 25, and (c) 50 (real mapping used). It can be noticed that the average distance between the gene feature and the features of the moving windows increases with AR model order, and it is minimal (zero) at the position of the gene.

The analysis was also performed using the real mapping rule. For a particular AR model of order p, the centroid of all coding region feature vectors was calculated, and the Euclidean distance of the feature vectors from the centroid was computed. The distances were similarly computed for noncoding region features from their centroid as well. The distribution density of these distance measures was obtained. The process was repeated for increasing model orders. The distributions from the coding region and noncoding regions were then compared using the Kolmogorov-Smirnov test [44]. Figure 10 shows the distribution densities for *S. cerevisiae* coding and noncoding regions for AR model orders 15 and 35, using binary SW mapping. The distribution densities obtained by using real-number mapping are depicted in Figure 11. Both coding and noncoding features are concentrated near their respective centroids. The noncoding features appear to be more concentrated around their centroid than the coding features.

The p values from the Kolmogorov-Smirnov test of the distributions of the coding and noncoding features using binary SW and real-number mapping, are shown in Figure 12. It is observed that the threshold $p = 0.05$ used in the hypothesis testing is achieved with an AR model order of 21 for the binary SW mapping and only 16 for the real mapping. Thus, it appears that such distance distributions can be used to further classify a DNA segment as coding or noncoding. It also appears that the real mapping is more effective than the binary SW mapping in this analysis.

6. CONCLUSION

A brief survey of the research on the analysis of DNA sequences from a signal processing perspective was presented. The use of nonparametric classical DSP tools like Fourier transforms and time-frequency analysis have been effective in studying DNA sequences of coding and noncoding regions. The use of parametric spectral analysis to capture certain spectral characteristics of such DNA regions was herein introduced. We applied the AR spectral analysis tools to analyze DNA sequences.

The analyses were of two basic types. First, the AR model parameters of the analyzed DNA segments were used to perform linear prediction analysis. The residual error was subsequently used to compare the analyzed segments. An observation of particular interest was that the AR model was very specific to the coding DNA sequences. This specificity increased with increasing model orders. Though the residual error analysis methodology could be used to compare AR models of different DNA segments, it was found not to be adequate for the characterization of these sequences. The AR

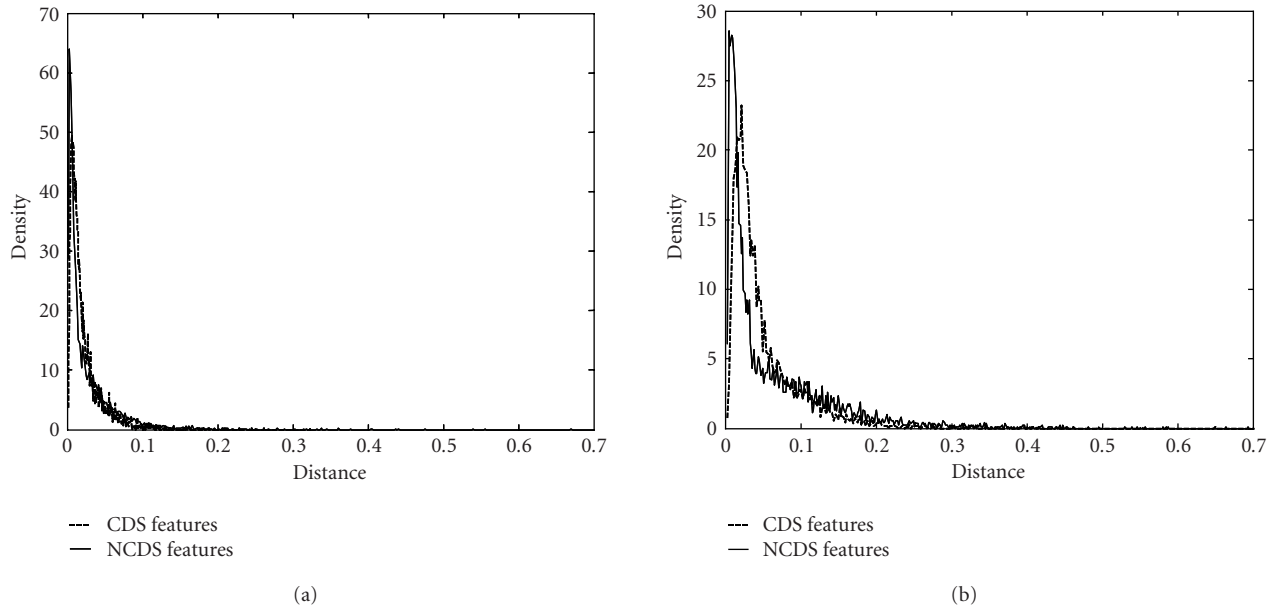


Figure 10: Distribution density of distances of coding segment (CDS) AR feature vectors and noncoding segment (NCDS) AR feature vectors from their respective centroids for AR model orders (a) 15 and (b) 35 (binary SW mapping used).

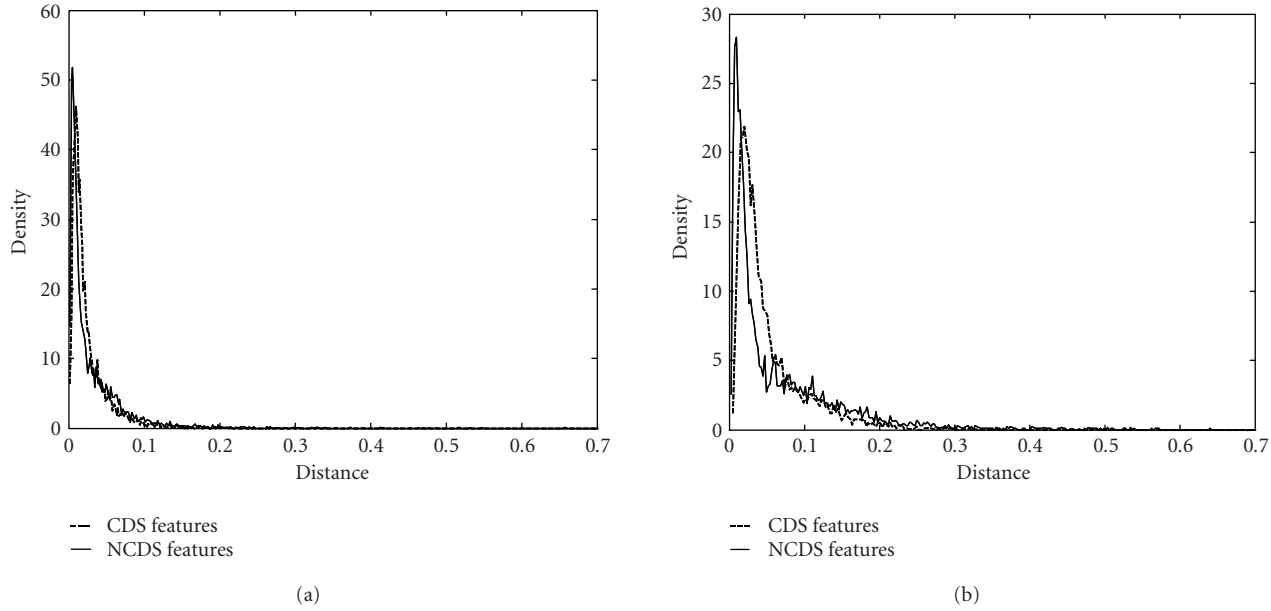


Figure 11: Distribution density of distances of coding segment (CDS) AR feature vectors and noncoding segment (NCDS) AR feature vectors from their respective centroids for AR model orders (a) 15 and (b) 35 (real mapping used).

model parameters themselves were then used as features for DNA string searches.

Depending on the type of the numerical mapping rule used, the AR feature-based string searching technique was highly effective in identifying all repeats of the query string, along with the locations of its complementary sequence. It was also possible to locate regions with similar chemical structures, for example, sequences of similar strong and weak hydrogen bonds. Thus different mapping rules can be used depending on the objective of the analysis. For example, the use of SW or RY mapping rules was necessary to locate regions of similar strong-weak hydrogen bonds or purine-pyrimidine structure. It was observed that modeling with a low-order AR model and working in the generated feature space was sufficient to locate the occurrence of complete genes in a long DNA sequence. Further analysis of the

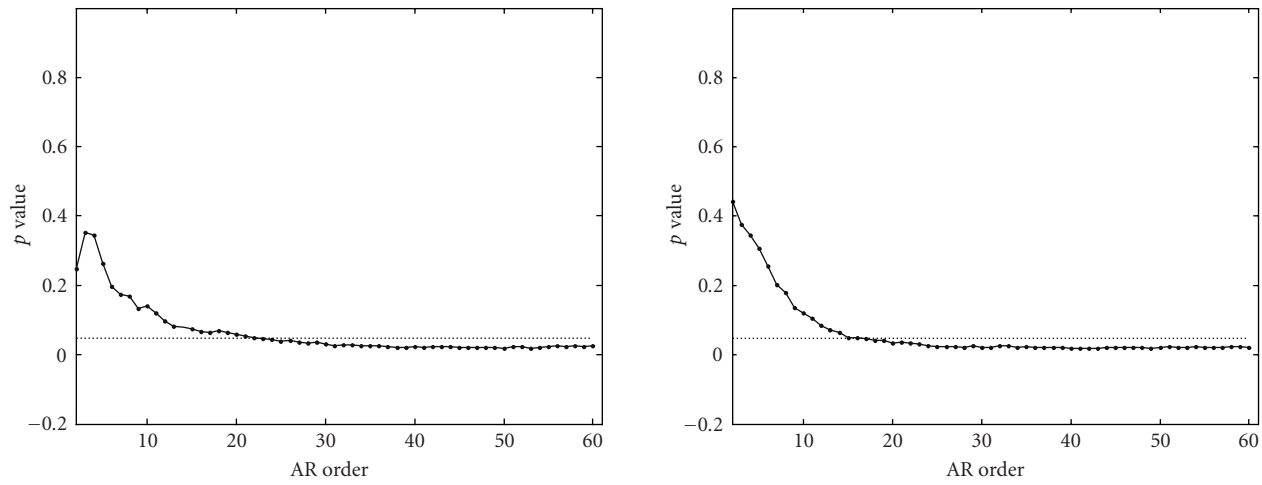


FIGURE 12: p values obtained from the Kolmogorov-Smirnov test, comparing the distribution of coding and noncoding AR features for (a) binary SW mapping and (b) real mapping. The 5% threshold used in the hypothesis testing is also plotted as a dotted horizontal line.

distribution of the coding and noncoding AR features revealed that these distributions differed significantly for high-dimension AR features. It would be of great interest to further investigate the biological implications of differences in the distributions of coding and noncoding region AR features.

The proposed analytical scheme can also be used for the analysis of other biochemical molecules, in addition to DNA, such as amino acid sequences. Further, like in speech recognition, AR features and their derivatives, such as cepstral features, could also be incorporated in an HMM-based gene-finding tool. Analysis of more genomic sequences along the lines proposed herein is underway.

ACKNOWLEDGMENTS

This work is partially supported by the National Institutes of Health through a Bioengineering Research Partnership Grant NS39687 to Dr. L. D. Iasemidis. Portions of the educational components of this work have been supported by the National Science Foundation Grant NSF0089075 to Dr. A. Spanias.

REFERENCES

[1] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya, and R. Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences," *Computer Applications in the Biosciences*, vol. 13, no. 3, pp. 263–270, 1997.

[2] D. Anastassiou, "Genomic signal processing," *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 8–10, 2001.

[3] B. D. Silverman and R. Linsker, "A measure of DNA periodicity," *Journal of Theoretical Biology*, vol. 118, pp. 295–300, 1986.

[4] P. P. Vaidyanathan and B.-J. Yoon, "Gene and exon prediction using allpass-based filters," in *Proc. Workshop on Genomic Signal Processing and Statistics (GENSIPS '02)*, Raleigh, NC, USA, October 2002.

[5] H. Herzel and I. Grosse, "Measuring correlations in symbol sequences," *Physica A*, vol. 216, no. 4, pp. 518–542, 1995.

[6] R. F. Voss, "Evolution of long-range fractal correlations and 1/f noise in DNA base sequences," *Phys. Rev. Lett.*, vol. 68, no. 25, pp. 3805–3808, 1992.

[7] S. V. Buldyrev, A. L. Goldberger, S. Havlin, et al., "Long-range correlation properties of coding and noncoding DNA sequences: GenBank analysis," *Phys. Rev. E*, vol. 51, no. 5, pp. 5084–5091, 1995.

[8] P. Bernaola-Galván, P. Carpena, R. Román-Roldán, and J. L. Oliver, "Study of statistical correlations in DNA sequences," *Gene*, vol. 300, no. 1-2, pp. 105–115, 2002.

[9] O. Weiss and H. Herzel, "Correlations in protein sequences and property codes," *Journal of Theoretical Biology*, vol. 190, no. 4, pp. 341–353, 1998.

[10] C. K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of DNA nucleotides," *Phys. Rev. E*, vol. 49, no. 2, pp. 1685–1689, 1994.

[11] W. Li, T. Marr, and K. Kaneko, "Understanding long-range correlations in DNA sequences," *Physica D*, vol. 75, no. 1–3, pp. 392–416, 1994.

[12] H. Herzel and I. Grosse, "Correlations in DNA sequences: The role of protein coding segments," *Phys. Rev. E*, vol. 55, no. 1, pp. 800–810, 1997.

[13] H. Herzel, E. N. Trifonov, O. Weiss, and I. Grosse, "Interpreting correlations in biosequences," *Physica A*, vol. 249, no. 1–4, pp. 449–459, 1998.

[14] W. Li, "The study of correlation structures of DNA sequences: a critical review," *Computers & Chemistry*, vol. 21, no. 4, pp. 257–272, 1997.

[15] L. Luo, W. Lee, L. Jia, F. Ji, and L. Tsai, "Statistical correlation of nucleotides in a DNA sequence," *Phys. Rev. E*, vol. 58, no. 1, pp. 861–871, 1998.

[16] D. Holste, I. Grosse, and H. Herzel, "Statistical analysis of the DNA sequence of human chromosome 22," *Phys. Rev. E*, vol. 64, no. 4, pp. 1–9, 2001.

[17] A. K. Mohanty and A. V. S. S. Narayana Rao, "Long range correlations in DNA sequences," preprint, 2002, http://arXiv.org/abs/physics/0202075.

[18] B. Audit, C. Thermes, C. Vaillant, Y. d'Aubenton-Carafa, J. F. Muzy, and A. Arneodo, "Long-range correlations in genomic

DNA: a signature of the nucleosomal structure," *Phys. Rev. Lett.*, vol. 86, no. 11, pp. 2471–2474, 2001.
[19] L. S. Marple, *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.
[20] B. Alberts, D. Bray, A. Johnson, et al., *Essential Cell Biology*, Garland Publishing, NY, USA, 1998.
[21] J. W. Fickett, "Recognition of protein coding regions in DNA sequences," *Nucleic Acids Research*, vol. 10, no. 17, pp. 5303–5318, 1982.
[22] J. W. Fickett, "The gene identification problem: an overview for developers," *Computers & Chemistry*, vol. 20, no. 1, pp. 103–118, 1996.
[23] R. Román-Roldán, P. Bernaola-Galván, and J. L. Oliver, "Sequence compositional complexity of DNA through an entropic segmentation method," *Phys. Rev. Lett.*, vol. 80, no. 6, pp. 1344–1347, 1998.
[24] J. A. Berger, S. K. Mitra, M. Carli, and A. Neri, "New approaches to genome sequence analysis based on digital signal processing," in *Proc. Workshop on Genomic Signal Processing and Statistics (GENSIPS '02)*, Raleigh, NC, USA, October 2002.
[25] I. Grosse, H. Herzel, S. V. Buldyrev, and H. E. Stanley, "Species independence of mutual information in coding and noncoding DNA," *Phys. Rev. E*, vol. 61, no. 5, pp. 5624–5629, 2000.
[26] J. W. Fickett and C. S. Tung, "Assessment of protein coding measures," *Nucleic Acids Research*, vol. 20, no. 24, pp. 6441–6450, 1992.
[27] P. Bernaola-Galván, I. Grosse, P. Carpena, J. L. Oliver, R. Román-Roldán, and H. E. Stanley, "Finding borders between coding and noncoding DNA regions by an entropic segmentation method," *Phys. Rev. Lett.*, vol. 85, no. 6, pp. 1342–1345, 2000.
[28] I. Grosse, P. Bernaola-Galván, P. Carpena, R. Román-Roldán, J. L. Oliver, and H. E. Stanley, "Analysis of symbolic sequences using the Jensen-Shannon divergence," *Phys. Rev. E*, vol. 65, pp. 041905-1–041905-16, 2002.
[29] M. Crochemore and R. Vérin, "Zones of low entropy in genomic sequences," *Computers & Chemistry*, vol. 23, no. 3-4, pp. 275–282, 1999.
[30] E. E. May, M. A. Vouk, D. L. Bitzer, and D. I. Rosnick, "A coding theory framework for genetic sequence analysis," in *Proc. Workshop on Genomic Signal Processing and Statistics (GENSIPS '02)*, Raleigh, NC, USA, October 2002.
[31] H. P. Yockey, "An application of information theory to the central dogma and the sequence hypothesis," *Journal of Theoretical Biology*, vol. 46, pp. 369–406, 1974.
[32] H. P. Yockey, *Information Theory and Molecular Biology*, Cambridge University Press, Cambridge, UK, 1992.
[33] A. A. Tsonis, J. B. Elsner, and P. A. Tsonis, "Periodicity in DNA coding sequences: implications in gene evolution," *Journal of Theoretical Biology*, vol. 151, pp. 323–331, 1991.
[34] P. D. Cristea, "Analysis of chromosome genomic signals," in *Proc. 7th International Symposium on Signal Processing and Its Applications (ISSPA '03)*, vol. 2, pp. 49–52, Paris, France, July 2003.
[35] D. H. Johnson and W. Wang, "Symbolic signal processing," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '99)*, pp. 1361–1364, Phoenix, Ariz, USA, March 1999.
[36] W. Wang and D. H. Johnson, "Computing linear transforms of symbolic signals," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 50, no. 3, pp. 628–634, 2002.
[37] D. S. Stoffer, D. E. Tyler, and A. J. McDougall, "Spectral analysis for categorical time series: Scaling and the spectral envelope," *Biometrika*, vol. 80, no. 3, pp. 611–622, 1993.
[38] D. S. Stoffer, D. E. Tyler, and D. A. Wendt, "The spectral envelope and its applications," *Statistical Science*, vol. 15, no. 3, pp. 224–253, 2000.
[39] A. Arneodo, E. Bacry, P. V. Graves, and J. F. Muzy, "Characterizing long-range correlations in DNA sequences from wavelet analysis," *Phys. Rev. Lett.*, vol. 74, no. 16, pp. 3293–3296, 1995.
[40] K. Bloch and G. R. Arce, "Time-frequency analysis of protein sequence data," in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP '01)*, Baltimore, Md, USA, June 2001.
[41] J. Song, T. Ware, and S.-L. Liu, "Test of origin site (oriC) and terminus (terC) of replication by wavelet analysis in bacteria," in *Proc. Workshop on Genomic Signal Processing and Statistics (GENSIPS '02)*, Raleigh, NC, USA, October 2002.
[42] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.
[43] J. D. Markel and A. H. Gray Jr., *Linear Prediction of Speech*, Springer-Verlag, NY, USA, 1976.
[44] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall/CRC Press, Boca Raton, Fla, USA, 2nd edition, 2000.

Niranjan Chakravarthy received the B.E. degree in electronics and communication engineering from the Government College of Engineering, Tamil Nadu, India, in 2001 and the M.S. degree in electrical engineering from the Arizona State University in 2003. He is currently working towards the Ph.D. degree with the Department of Electrical Engineering at Arizona State University. He is a Research Assistant at the Digital Signal Processing and Brain Dynamics Laboratories and currently pursues research on the prediction and control of epileptic seizures and genomic signal processing. His research interests include digital signal processing, time series modeling, and systems theory and applications to physical and biological systems.

A. Spanias is a Professor of electrical engineering at Fulton School of Engineering, Arizona State University. His research interests are in adaptive signal processing and speech processing. He received the 2003 Teaching Award from the IEEE Phoenix Section for the development of J-DSP. He is a member of the IEEE-CAS Society DSP Technical Committee and has served as a Member in the Technical Committee on Statistical Signal and Array Processing of the IEEE Signal Processing Society (SPS). He has served as an Associate Editor of the IEEE Transactions on Signal Processing, General Cochair of the 1999 International Conference on Acoustics Speech and Signal Processing (Phoenix), IEEE Signal Processing Vice President for Conferences, and Chair of the Conference Board. He served as a Member in the IEEE Signal Processing Executive Committee and as an Associate Editor of IEEE Signal Processing Letters. He is currently serving as a Member in the IEEE SPS Publications Board, and Member-at-Large of the IEEE SPS Conference Board. He has been Chair of the Phoenix IEEE Communications and Signal Processing Chapter, and is a Member in Eta Kappa Nu and Sigma Xi. Andreas Spanias is corecipient of the 2002 IEEE Donald G. FinkPaper Award, and was recently elected as a Fellow of the IEEE. He is appointed as 2004 Distinguished Lecturer of the IEEE SPS.

L. D. Iasemidis received the Diploma in electrical and electronics engineering from the National Technical University of Athens in 1982, M.S. in Physics, M.S. and Ph.D. in biomedical engineering from the University of Michigan, Ann Arbor, Mich in 1985, 1986, and 1991, respectively. Dr. Iasemidis is currently an Associate Professor of Bioengineering at the Arizona State University, Tempe, Ariz, and Director and Founder of the ASU Brain Dynamics Laboratory. Dr. Iasemidis is recognized as an expert in dynamics of epileptic seizures, and his research and publications have stimulated an international interest in the prediction and control of epileptic seizures, and understanding of the mechanisms of epileptogenesis. He is currently on the Editorial Board of Epilepsia and IEEE Transactions on Biomedical Engineering, and is a Reviewer of NIH. He has reviewed articles for more than 10 scientific journals. His research interests are in the areas of biomedical and genomic signal processing, complex systems theory and nonlinear dynamics, neurophysiology, monitoring and analysis of the electrical and magnetic activity of the brain in epilepsy and other brain dynamical disorders, intervention and control of the CNS, neuroplasticity, rehabilitation, and neuroprosthesis. Dr. Iasemidis' research has been funded by NIH, VA, DARPA and the Whitaker Foundation.

K. Tsakalis received his Ph.D. degree in electrical engineering from the University of Southern California. He is currently a Professor of electrical engineering at Arizona State University. His interests are in robust adaptive control, time varying systems, applications of control, identification, and optimization in semiconductor manufacturing problems, and, more recently, the application of adaptive systems theory on the prediction and control of epileptic seizures.

EURASIP Journal on Applied Signal Processing 2004:1, 29–42

Spectrogram Analysis of Genomes

David Sussillo
Department of Electrical Engineering, Columbia University, NY 10027, USA
Email: sussillo@ee.columbia.edu

Anshul Kundaje
Department of Electrical Engineering, Columbia University, NY 10027, USA
Email: abk2001@cs.columbia.edu

Dimitris Anastassiou
Department of Electrical Engineering, Center for Computational Biology and Bioinformatics (C2B2) and Columbia Genome Center, Columbia University, NY 10027, USA
Email: anastas@ee.columbia.edu

Received 28 February 2003; Revised 22 July 2003

We perform frequency-domain analysis in the genomes of various organisms using tricolor spectrograms, identifying several types of distinct visual patterns characterizing specific DNA regions. We relate patterns and their frequency characteristics to the sequence characteristics of the DNA. At times, the spectrogram patterns can be related to the structure of the corresponding protein region by using various public databases such as GenBank. Some patterns are explained from the biological nature of the corresponding regions, which relate to chromosome structure and protein coding, and some patterns have yet unknown biological significance. We found biologically meaningful patterns, on the scale of millions of base pairs, to a few hundred base pairs. Chromosome-wide patterns include periodicities ranging from 2 to 300. The color of the spectrogram depends on the nucleotide content at specific frequencies, and therefore can be used as a local indicator of CG content and other measures of relative base content. Several smaller-scale patterns are found to represent different types of domains made up of various tandem repeats.

Keywords and phrases: DNA spectrograms, frequency-domain analysis, genome analysis.

1. INTRODUCTION

Color spectrograms of biomolecular sequences were introduced in [1, 2] as visualization tools providing information about the local nature of DNA stretches. These spectrograms give a simultaneous view of the local frequency throughout the nucleotide sequence, as well as the local nucleotide content indicated by the color of the spectrogram. They are helpful not only for the identification of genes and other regions of known biological significance, but also for the discovery of yet unknown regions of potential significance, characterized by distinct visual patterns in the spectrogram that are not easily detectable by character string analysis. Further, they have been found to give global information about whole chromosomes as well.

In this paper, we discuss the features and patterns that such spectrograms reveal. We applied a slightly modified version (described below) of the spectrogram development tool introduced in [1, 2] that provides a more direct manifestation of the local relative nucleotide content in the color of the spectrogram, and explored the patterns characteristic in the genomes of various organisms. We created color spectrograms of various frequency bandwidths and sequence lengths. Although the genomes of these organisms vary greatly in size, chromosome number, and complexity, we found many interesting features, some of which are common to all organisms and some are unique to a particular organism. Some of the uncovered patterns relate to the overall chromosome structure or to protein coding. On some occasions, the specific function of a protein could be understood by visual comparison to other proteins.

We analyzed some parts of the genomes from *E. coli*, *M. tuberculosis*, *S. cerevisiae*, *P. falciparum*, *C. elegans*, *D. melanogaster*, and *H. sapiens*, viewing chromosomes and chromosome subsequences using the tricolor spectrogram with as much or as little frequency and sequence resolution as necessary. We allowed zooming in and out in both the frequency and sequence dimensions, thus facilitating easy navigation of DNA that is normally intimidating in its complexity. A set of colors was initially chosen for the four different bases to maximize the discriminatory power of the spectrogram. Depending on the pattern, we adjusted the frequency

and sequence resolutions so that the prominent frequencies were accurately highlighted and thus we were able to view different features of the chromosome with great precision. When possible, we referenced the subsequence from which the pattern was created with various public databases to further ascertain the function of the region. We then annotated the patterns with the type of pattern, prominent periodicities, position in the chromosomal DNA sequence, and corresponding position in the protein sequence if the DNA was coding. Thus, we related pattern shape and color to significant structural and functional elements in the genome. Most of our searches were exhaustive, and the patterns shown in this paper are exemplary of myriad patterns in the various genomes.

The spectrograms were developed using the short-time Fourier transform, that is, by applying the N-point discrete Fourier transform (DFT) over a sliding window of size N. The difficulty in creating DNA spectrograms results from the fact that DNA sequences are defined by character strings rather than numerical sequences. This problem can be solved by considering the *binary indicator sequences* $u_A[n]$, $u_T[n]$, $u_C[n]$, and $u_G[n]$, taking the value of either one or zero depending on whether or not the corresponding character exists at location n. These four sequences form a redundant set because they add to 1 for all n. Therefore, any three of these sequences are sufficient to determine the character string. In [1, 2], color spectrograms are defined by creating RGB superposition, using the colors red, green, and blue, of the spectrograms for the numerical sequences

$$\begin{aligned} x_r[n] &= a_r u_A[n] + t_r u_T[n] + c_r u_C[n] + g_r u_G[n], \\ x_g[n] &= a_g u_A[n] + t_g u_T[n] + c_g u_C[n] + g_g u_G[n], \\ x_b[n] &= a_b u_A[n] + t_b u_T[n] + c_b u_C[n] + g_b u_G[n], \end{aligned} \quad (1)$$

in which, to enhance the discriminating power of the visualization, the coefficients in the above equations are chosen by assigning each of the four letters to a vertex of a regular tetrahedron in the three-dimensional space. In the present implementation, we further improve the discriminating power by ensuring that all points in the tetrahedron have different absolute values with respect to any axis using the following choice of coefficients:

$$\begin{array}{lll} a_r = 0, & a_g = 0, & a_b = 1, \\ t_r = 0.911, & t_g = -0.244, & t_b = -0.333, \\ c_r = 0.244, & c_g = 0.911, & c_b = -0.333, \\ g_r = -0.817, & g_g = -0.471, & g_b = -0.471. \end{array} \quad (2)$$

To illustrate, we first consider three examples that demonstrate both the use of color and periodicity in the spectrogram. The horizontal axis indicates the location in the DNA sequence measured in base pairs (bp) from the origin and the vertical axis indicates the discrete frequency of the DFT measured in cycles per STFT window size. The corresponding period is equal to N/k, where k is the discrete frequency and N is the STFT window size.

Unlike the traditional spectrograms that employ pseudocolor to achieve greater contrast, the spectrograms that are used to visualize DNA sequences contain useful information

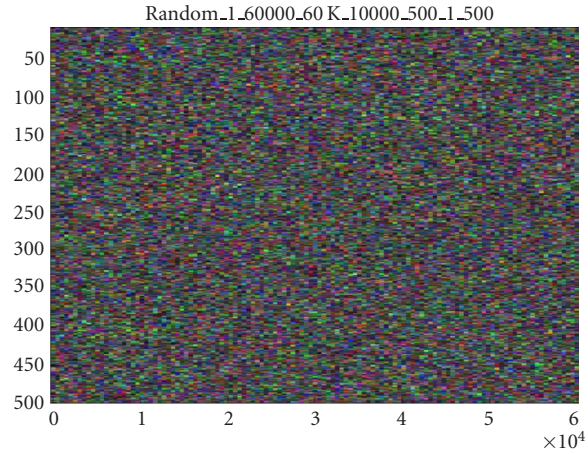


Figure 1: Spectrogram of a random DNA sequence of length 60 kbp. No obvious patterns are discernable. Spectrogram titles are annotated with a helpful name or accession tag, sequence-start index, sequence-end index, approximate sequence length, DFT window size, window overlap, lowest frequency shown in image, and highest frequency shown in image.

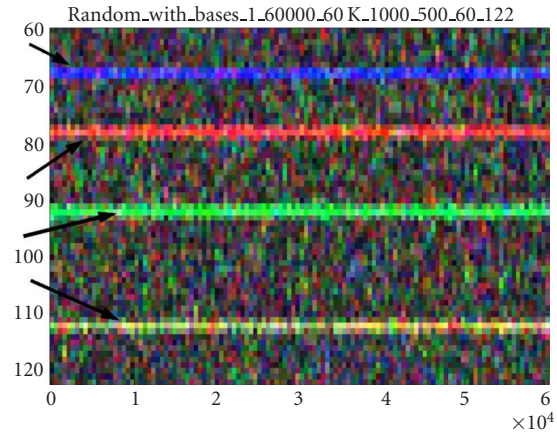


Figure 2: Spectrogram of random DNA of length 60 kbp with bases A, T, C, and G with periods 15, 13, 11, and 9, respectively. The nucleotide A is represented by the color blue, T by red, C by green, and G by yellow. Arrows mark the different periodicities.

encoded in color. The colors for the nucleotides A, T, C, and G are blue, red, green, and yellow, respectively. These colors were chosen to optimize the discrimination between different nucleotides. As a rule of thumb, the interaction between the various nucleotides is visualized as the superposition of colors representing those nucleotides. Thus, a sequence composed of ATATAT ... would have a purple bar at the frequency corresponding to period 2. The first spectrogram (Figure 1) shows a spectrogram created from a sequence of 60000 "totally random" nucleotides. The sequence was created from an independent identically and uniformly distributed random sequence model so that every position has equal chance of being an A, T, C, or G. No obvious patterns are noticeable. The second spectrogram (Figure 2) shows the same sequence as the first but with a modification

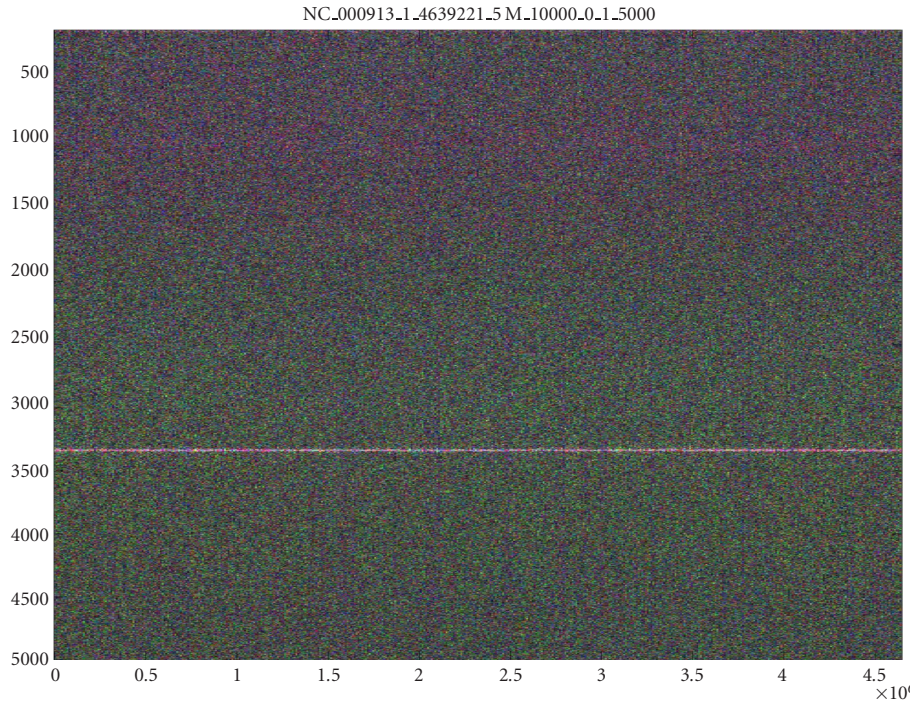


Figure 3: Spectrogram of the entire *E. coli* K12 chromosome (about 4.6 Mbp). The line marking the 3-base periodicity of protein-coding regions extends without a visible break across the entire chromosome. There is a change in color going from higher frequencies (greenish) to lower frequencies (purplish).

so that every 15 nucleotides, there is an *A*; every 13 nucleotides, there is a *T*; every 11 nucleotides, there is a *C*; and every 9 nucleotides, there is a *G*. This figure demonstrates that even in complicated sequences, *A* is mapped by the color blue, *T* by red, *C* by green, and *G* by yellow.

2. CHROMOSOME-WIDE PATTERNS

Distinguishing patterns by their size makes a simple categorization. Those patterns composed of millions of bp are considered large; those that are composed of up to several hundred thousand nucleotides are medium; and those patterns consisting of up to several thousand bp are small. Typically, larger patterns represent structural elements and smaller patterns are useful in visualizing something about a protein-coding region. Here, we focus first on large patterns. In doing so, we focus on the general characteristics of the chromosome-wide spectrogram.

2.1. *E. coli*

Figure 3 shows the spectrogram of the entire chromosome for the bacteria *E. coli* using STFT window size $N = 10\,000$. The count among all nucleotides in *E. coli* is roughly equal (A=1142136, T=1140877, C=1179433, G=1176775) and the total number of nucleotides is over 4.6 Mbp. The most salient feature is the strong intensity with periodicity 3 (frequency 3333) that corresponds to protein-coding regions. The fact that protein-coding regions in DNA typically have a peak at the frequency of 3 periodicity in their Fourier spectra is well known [3, 4, 5, 6]. The whiteness of this line shows that most of the bases are being used in protein coding, and this is clearly reflected by the continuity and intensity of the line with periodicity 3. Second, at regular intervals along the DNA sequence, there appear thin veins of purple, implying AT rich areas intermittently placed along chromosome. Finally, there is a general shift in hue as the frequency decreases. The larger frequencies are more greenish in hue and the lower frequencies are more purplish. The purplish hue extends over from about the 6.5-base periodicity and upwards and shows that even while apparently coding for genes almost everywhere on the chromosome, the chromosome is also preserving higher periodicities involving the nucleotides *A* and *T*. This is particularly interesting considering that the total number of each of the four bases in the genome is nearly equal. The purplish hue in the lower frequencies may be related to the twisting of the DNA molecule that leads to helical repeats.

2.2. *C. elegans chromosome III*

We now turn our attention to the multicellular organism *C. elegans*. Figure 4 shows the DNA spectrogram of chromosome III. The general hue of the spectrogram is darker than that of *E. coli*. This relates directly to the relative number of bases in chromosome III (A=4444502, T=4423430, C=2449072, G=2466240). The horizontal line of intensity marking the 3-base periodicity is much less pronounced than *E. coli* in that there are more gaps along the sequence. This is consistent with the general rule that eucaryotic DNA

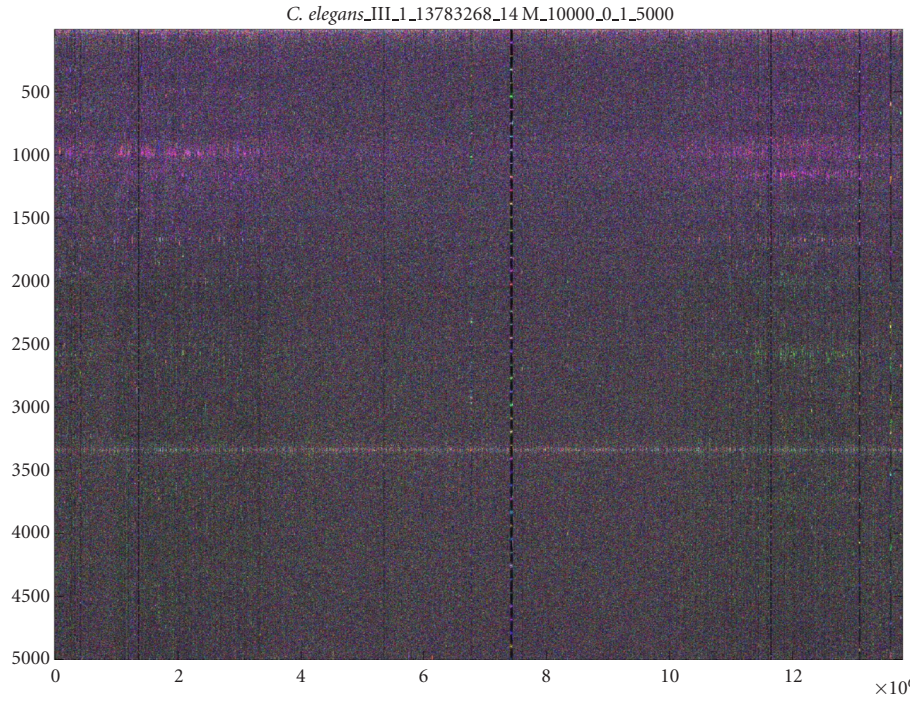


Figure 4: Spectrogram of the chromosome III of *C. elegans* (13.8 Mbp). The 3-base periodicity relating to protein coding is noted. A minisatellite is noticeable at 7.4 Mbp (see Figure 16). Various periodicities are noticeable, in particular, the purple 10+-base periodicity in both chromosome arms and coincident 8, 9-base and green 3.8-base periodicities in the right chromosome arms.

contains more noncoding DNA such as intergenic DNA and introns. In the middle of the spectrogram, there is a vertical bar that identifies a "minisatellite," roughly 50 kbp in length. The details of minisatellites are explained in Section 3.1. On some regions, there are strong horizontal bands of intensity between the frequencies representing the 8-base periodicity and 9-base periodicity (at 8.7) and also just above 10 (at 10.2, which we call the "10+ periodicity") throughout the entire chromosome. In the right part of the spectrogram, (close to 12 Mbp) there are strong periodicities involving the color green and thus the bases GC at 3.9.

The 10+ periodicity appears to be of special importance. Figure 5 shows the magnitude plot of the DFT for the four nucleotides in the subsequence 1456174 – 1596391. Each separate base is plotted with a different color. The frequency range shown corresponds to periods 8 through 12. The periodicities at 10+ are the strongest in the bases *A* & *T* (area indicated by arrow). This periodicity may relate to DNA helical structure, which has a periodicity of 10.4 bp on average [7, 8, 9, 10]. The 10+ periodicity may also be related to folding around nucleosomes, as the nucleotides *A* and *T* are preferred in the minor grove when binding to the nucleosome core. The DNA double helix kinks when wrapped around the nucleosome core, thus reducing its helical periodicity to 10.39±0.02 bp [9]. We found that the maximal intensity of this band has a 10.2-base periodicity.

We further searched chromosome III of *C. elegans* at much lower frequencies and found a 1.5 Mbp long (0.8 Mbp–2.6 Mbp subsequence) bubble centered on period 300. This was accomplished using a DFT window size of 40000. Figure 6 shows this spectrogram with the two bubbles centered at period 300 marked by arrows. This was the only example of a periodicity found around 300 and it is unclear what biological significance the bubble may have. Figure 7

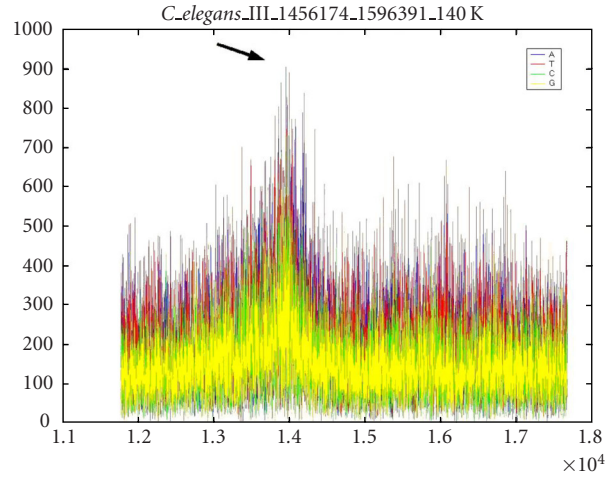


Figure 5: DFT magnitude plot of 140 Kbp section of *C. elegans* chromosome III showing higher values at period 10+ in all bases, but particularly *A* and *T*. An arrow marks the peak in the periodicity range of 9.9–10.5.

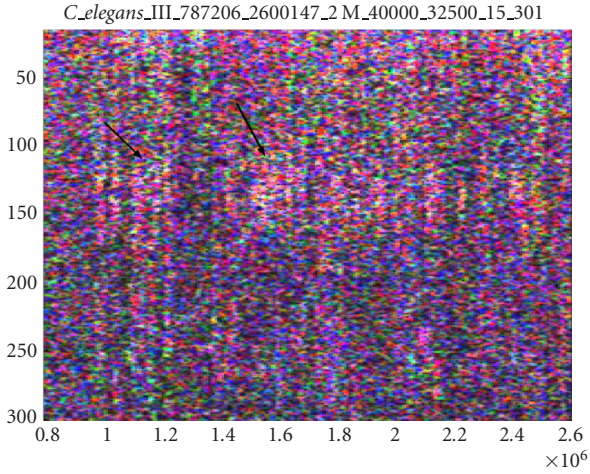


Figure 6: Spectrogram showing an intensity increase around a periodicity of 300 in *C. elegans* chromosome III. The sequence is roughly 2 Mbp in length. Arrows mark two such areas.

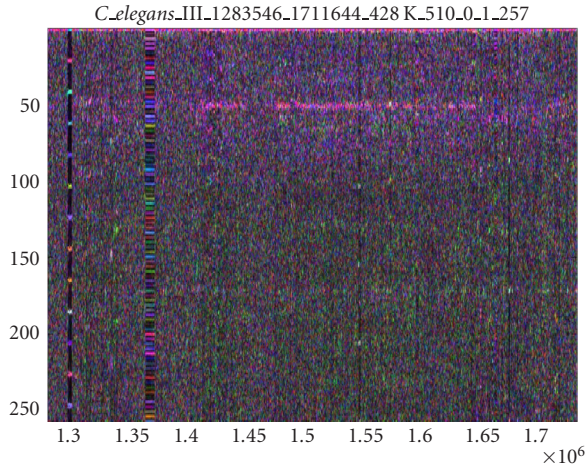


Figure 7: Spectrogram showing a strong coincident 10+-base periodicity in the same DNA sequence shown in Figure 6 (coincident with 300-base periodicity). This spectrogram corresponds to the rightmost arrow in Figure 6 and is 428 Kbp in length.

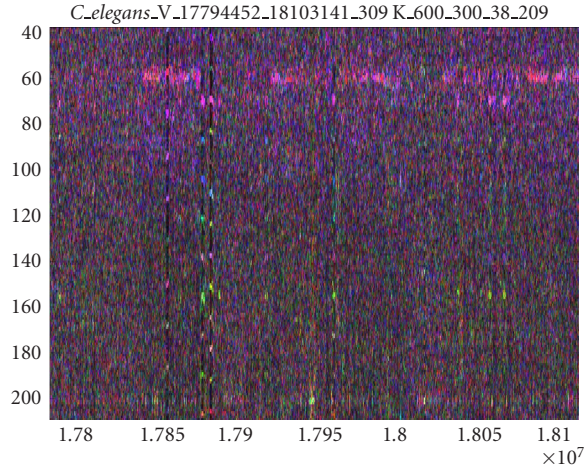


Figure 8: Spectrogram showing antagonism between 10+-base and 3-base periodicities in *C. elegans* chromosome III (300 Kbp). The 10+-base periodicity is at the top of the figure while the 3-base periodicity is shown at the very bottom.

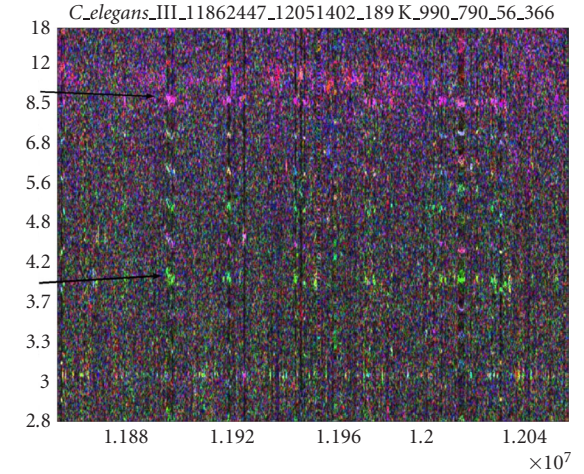


Figure 9: Spectrogram of 189 Kbp section of the right arm of *C. elegans* chromosome III. Note that the periodicity is shown on the vertical scale. The arrows point to sections of the spectrogram, showing a single instance of the highly dispersed repeat family. Variations of the pattern can be seen throughout the spectrogram. A purple 8.75-base periodicity, as well as a green 3.9-base periodicity, identifies this family of strings. The harmonics between 3.9 and 8.75 (the beads of color between 3.9 and 8.75) change color from one repeat to another, indicating that they are different but related strings. These tandem repeats are non-protein-coding regions. The 10+-base periodicity is antagonistic with the repeat family. This pattern is found over 3 Mbp of the right arm of the chromosome.

shows the same area of the chromosome (1.4 Mbp–1.6 Mbp) at higher frequency resolution, thus showing smaller periodicities. There appears to be coincident intensity at 10+ period in exactly the same area of intensity in the 300-period bubble.

In general, it appears that there are both "antagonism" and "cooperation" between various periodicities in all the chromosomes that we analyzed. For example, the arms of *C. elegans* chromosome III show obvious cooperation among many periodicities appearing simultaneously (Figure 7). Some cooperative periodicities are harmonics of a fundamental periodicity, indicating a repeat region (see Section 3.1). On the other hand, Figure 8, a subsection of chromosome V of *C. elegans*, shows an example of antagonism between the 3-base periodicity and the 10+-base periodicity. The brightest spots on the 3-base periodicity are the dimmest spots on 10+-base periodicity and vice versa. An explanation may be that in non-protein-coding regions, the periodicities due to structural constraints are more pronounced.

We identified a unique family of repeats in chromosome III via cooperation among periodicities. In the right arm of

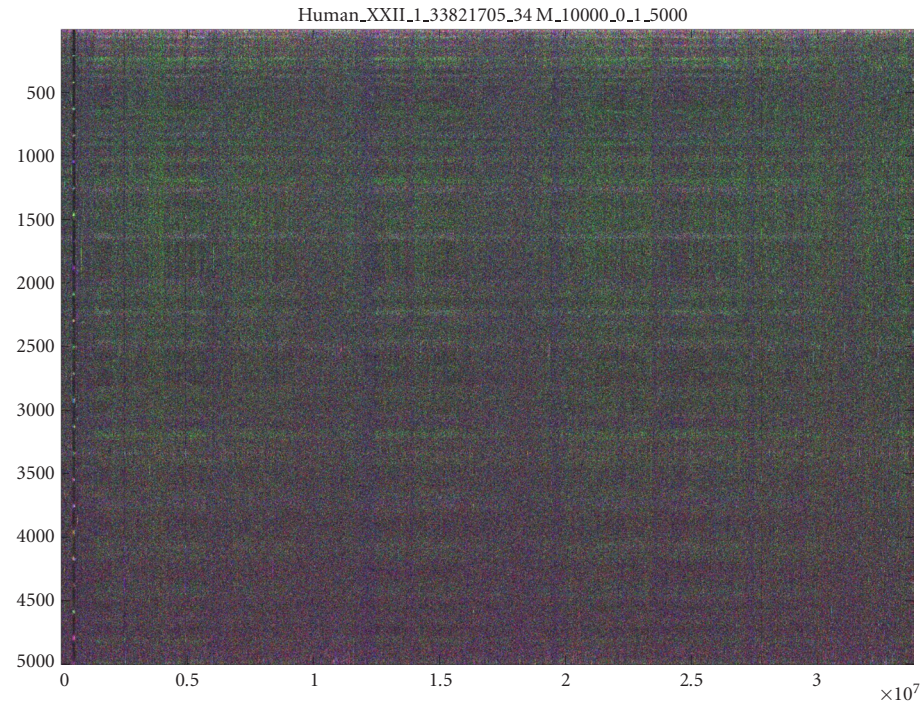


Figure 10: Spectrogram of human chromosome 22. Noticeably absent is the line representing the 3-base periodicity relating to protein coding. The 800 or so genes located on chromosome 22 simply do not cover enough of the chromosome to make a visible line at the resolution of 34 Mbp. Many periodicities are visible across the entire length of the chromosome.

chromosome III (10–13 Mbp), it appears that the AT rich 8.75-base periodicity almost always coincides with the GC-rich 3.9-base periodicity (Figure 4). In fact, the pattern found in the right arm of chromosome III, which shows cooperative periodicities at the chromosome level, is composed of a family of strings that are repeated in a very haphazard fashion. These strings are both heavily mutated and heavily dispersed throughout the chromosome. Yet throughout the many variations within the family, the 8.75-base and 3.9-base periodicities are always conserved. One instance of a repeat unit is "tttccggcaaattggcaagctgtcggaatttaaaa." Figure 9 shows how the family of strings manifests within the DNA. An instance of the family repeats for a hundred to a couple thousand bp, and these regions are interspersed among other DNA every 10 Kbp or so. Repeats of this family of mutated strings, unbelievably, are responsible for the macroscopic character of the right arm (3 Mbp region) of chromosome III (Figure 4). It is unclear whether or not the conserved periodicities imply a conserved biological function for the string, or whether it is simply a mathematical or biological property of this family of strings that certain of its periodicities are more easily preserved against mutation.

2.3. Human chromosome 22

The last full chromosome we analyzed was human chromosome 22. The actual sequence used was the correct reordering of contigs found in hs_chr22.fa from NCBI. This ordering is: NT_011516.5, NT_028395.1, NT_011519.9, NT_011520.8, NT_011521.1, NT_011522.3, NT_011523.8, NT_030872.1, NT_011525.4, NT_019197.3, and NT_011526.4. Figure 10 shows the 33 million-plus nucleotides of human chromosome 22. A strong bar of intensity representing the 3-base periodicity is strikingly absent. Closer inspection shows that there are many genes along chromosome 22 but they are far enough apart so that there is no noticeable band. There are around 30 easily noticeable, different periodicities that span the entire length of the chromosome. The biological function of these periodicities is unclear. Some periodicities may reflect higher periodicities in the form of harmonics.

The higher structures in DNA folding are unknown, so we were interested in determining whether or not spectrogram analysis would yield any insights into the DNA folding and superstructure. It is known that DNA has many orders of structure [11]. The simplest of such a superstructure is that of the nucleosome. Nucleosomes are an essential structural element in DNA: 146 bp wrap twice around a single nucleosome core particle, and between two nucleosomes, there is "linker" DNA that ranges in size but on the whole, nucleosomes repeat at intervals of about 200 bp. Nucleosome core particles will bind randomly along a sequence of DNA. However, AT rich sequences in the minor groove of DNA bind preferentially to the nucleosome core particle. Since euchromatic DNA is arranged in nucleosomes that require structural bending of the DNA, it is plausible that there might be some evidence of this structure in the form of a strong band with intensity of 200-base periodicity. We

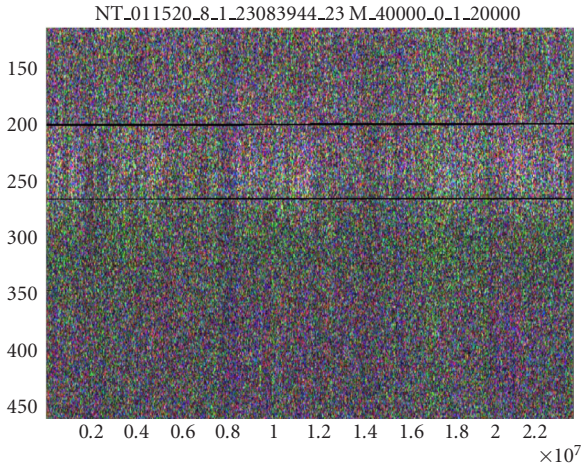


Figure 11: NT_011520.8 (23 Mbp in length) of human chromosome 22. The two artificial black lines mark the 150-base and 200-base periodicities. This band of intensity may relate to the folding of DNA into nucleosomes.

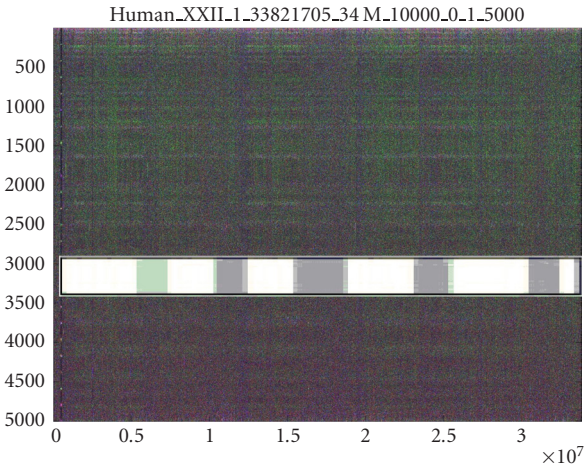


Figure 12: Spectrogram of human chromosome 22 matched up with a part of the Giemsa-stained schematic of the same chromosome. There is a visual agreement between AT-rich regions and dark bands of Giemsa staining.

viewed contig NT_011520.8 (23 Mbp in length) of chromosome 22 with a very large DFT window in order to get high-frequency resolution. Figure 11 shows contig NT_011520.8 in the frequency range to show the 200-base periodicity. Two dark lines mark the 150-base periodicity and the 200-base periodicity, indicating a band of increased intensity between these markers. This intensity band may represent periodicities involved in nucleosome-chromatin superstructure. This 150 − 200-base periodicity band was the only one found in our exploration of various chromosomes. The 150 − 200-base periodicity was the largest periodicity found in the human chromosome 22.

We found an interesting feature of human chromosome 22 in the variation of the CG versus AT rich regions. As mentioned earlier, the color of the DNA spectrogram reflects the ratio of different nucleotides in the sequence (Figures 1 and 2). Different genomes vary greatly in the percentages of nucleotides that compose the sequence. As shown in Figure 10, a single chromosome can have long expanses of a single distribution of bases. Figure 10 shows clear boundaries between areas of high CG content and areas with lower GC content. The laboratory technique of Giemsa staining is correlated to the relative content of CG nucleotides. The GC-rich regions of DNA are responsible for the light bands in Giemsa staining while GC-poor regions create the dark bands [12]. We matched up a schematic of human chromosome 22 marked by Giemsa staining with our DNA spectrogram and found a reasonable alignment between the dark bands of the Giemsa stained chromosome schematic and the darker, purplish AT regions of the spectrogram (Figure 12). The match was made by aligning the rightmost part of the spectrogram with the "bottom" of the chromosome, that is, contig NT_011526.4. Because the spectrogram encodes different colors for each different base, it is easy to get a feeling for the relative number of bases in a sequence.

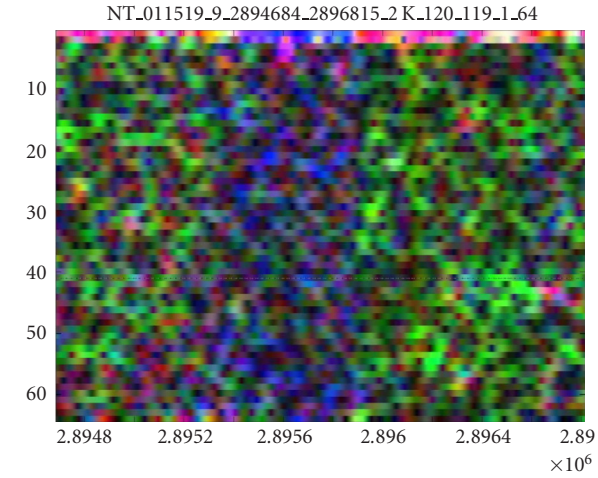


Figure 13: Spectrogram showing two CpG islands separated by a sequence very rich in the nucleotide *A*. Both islands yielded blast results showing T-box genes.

CpG islands [13] are DNA stretches in which a particular methylation process that normally reduces the occurrence of CG dinucleotides is suppressed, and therefore CG nucleotides appear more frequently than elsewhere. Such stretches are also readily identified using the DNA spectrogram. For example, we found two CpG islands simply by searching for the greenest subsequence we could locate in the genome. This simple color criterion yielded two CpG islands, shown in Figure 13. Figure 14 shows the results from the Emboss CpGplot program on the sequence that generated the spectrogram. The CpGplot figure shows that the CpG islands are located in exactly those sequences that are most green in the spectrogram. The subsequences from which the spectrogram was created were blasted on the NCBI website and both "green" sequences coded for T-box genes. The T-box genes

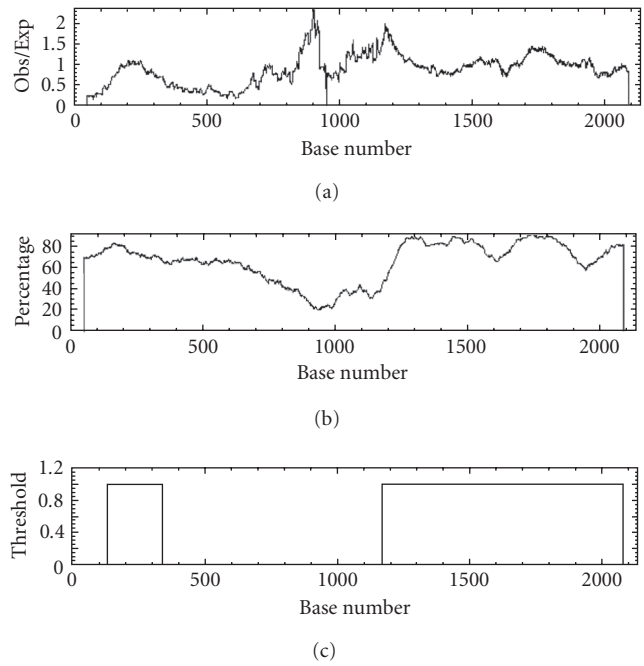


Figure 14: Graphs showing the results from the emboss CpGplot routine. (c) shows the predicted CpG islands (putative islands).

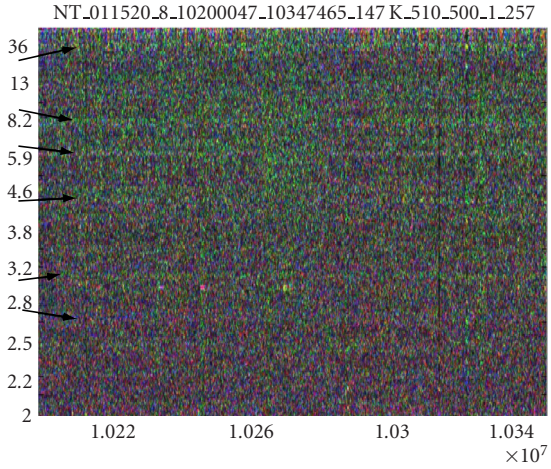


Figure 15: Spectrogram of a 147 Kbp section of human chromosome 22. Periodicity is shown on vertical scale. Contrasted with Figure 9, this spectrogram shows that the chromosome-wide periodicities found in human chromosome 22 are qualitatively different from those found in the right arm of C. elegans chromosome III. The periodicities here are much more finely embedded in the DNA and do not represent any obvious family of strings discretely interspersed throughout the region. Arrows point out some of the chromosome-wide periodicities found in Figure 10.

share a common binding domain, called the T-box. Finding this gene is in keeping with the idea that CpG islands encode for housekeeping genes.

Finally, we wondered whether or not the chromosome-wide periodicities found in human chromosome 22 are caused by a highly dispersed repeat family similar to that found in the right arm of *C. elegans* chromosome III. This appears not to be the case. The macroscopic appearance of periodicities in *C. elegans* is caused by widely placed repeats with such strong characteristics as shown at the macroscopic level. In the case of human chromosome 22, it appears as if the very fabric of intergenic DNA is woven with a string patterns that employs characteristic periodicities seen at the chromosome level (Figure 15). In other words, it appears as if the majority of intergenic DNA carries the periodicities found at the macroscopic level. Initial investigations show that these embedded periodicities are not found in chromosome 17 of the mouse.

3. SMALL PATTERNS

We now turn our attention to smaller subsequences of interest in various genomes. Color spectrograms can clearly identify, by their special signatures, several patterns including repetitive areas of biological significance such as particular triplet repeats [14], GATA repeats [15], or other characteristic repeating motifs in protein structures [16].

The sequences that we analyzed were typically several thousand bp in length, no more than a hundred thousand bp. The majority of smaller sequences we analyzed relates to protein-coding regions or repetitive sequences in non-protein-coding regions. The public databases were often helpful in matching spectrogram patterns to proteins. We annotated the spectrograms with the type of pattern, prominent periodicities, position in the chromosome, and corresponding position in the protein sequence if the DNA was coding.

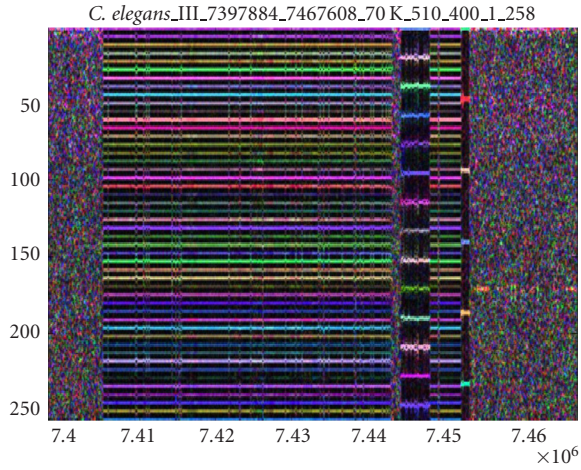


Figure 16: Spectrogram showing a minisatellite with repeat unit of length 95 bp in chromosome 3 of *C. elegans*. Slight variations in the basic repeat pattern can be seen as vertical lines that appear blurry. The minisatellite is interrupted by a small amount of nonrepeat DNA as well as an even simpler repeat unit of length 5 kbp.

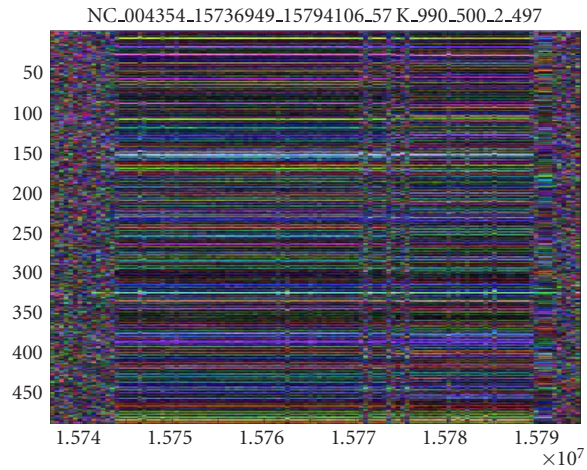


Figure 17: Spectrogram showing 40 kbp minisatellite in chromosome X of *D. melanogaster*. The repeat length is 298 bp. Three strong interruptions can be seen as vertical lines just right of the center.

We used a number of public databases during our analysis of DNA color spectrograms. The determination of whether or not a sequence was protein coding was accomplished using the SGD and GenBank databases. We also noted structural and functional details of the corresponding protein. Domains and motifs corresponding to the protein region were discovered using PFAM, CYGD, and SWISS-PROT databases for yeast, WormPD for *C. elegans*, and GenBank annotations for humans. Structural predictions were obtained using Pedant (CYGD) and GCG PepStruct (SGD). To test specifically the beta-helix supersecondary structure, the Betawrap program (Betawrap) was used.

At smaller length scales, the parameters of the STFT are very important in visualization; we initially experimented these parameters with different DFT window sizes for the spectrogram. It was found that using roughly 6 K nucleotides per spectrogram image with a DFT window size of 120 and an overlap of 119 gives the most optimal visualization of protein-coding regions. The choices of DFT window size and overlap were found to be particularly important in determining the pattern shape.

3.1. Minisatellites

The genome has repetitive regions varying in range from 500 bp to 100 kbp in length. These regions are composed of a smaller repeat unit that varies in length. If the length of the repeat unit is below 100, then the overall repeat region is called a minisatellite or variable number of tandem repeats (VNTR). Minisatellites have been found to vary in the number of tandem repeats in different germ cells and thus, make useful genetic markers [17]. A minisatellite composed of roughly 30 kbp was found in *C. elegans* chromosome III (Figure 16). It is also visible in the middle of Figure 4. The tandem repeat is composed of the 95 bp-long unit sequence "ttttgataattactgcctccagaaattgatgattttcccattgatttgtctacatagggca tcgaaaagcacccaatatttagagaacagaaga" and slight variants. According to "WormBase," this subsequence of chromosome III is completely unannotated. Another 40 kbp minisatellite was found in chromosome X of *D. melanogaster* (see Figure 17). The tandem repeat sequence is composed of the 298 bp-long unit sequence "ttcatttcaagaatccagtgcagaagaaaatcaaatgacagaa gtgccatggacactatcaacatcactttcccaatcaagttcaaaaacaaagaatatatttt tcgagtcaaagtgtaaatgaagacaacatttctcaagaagatacaaggacaccatcaa tatctgtcccacaatcaagtacaacagcaaatagattacttacaggttcgggtgcagaa gagccaacagctcaagaggagacatcggaactttcaaaatccttacctcaattaacaa cagaagagagcagttcattt." The GenBank file indicates that the location of the predicted gene CG32580 is in the region 15740143-15792683. Both minisatellites are large enough to be identifiable when viewed from a spectrogram of the entire chromosome.

Spectrogram visualization of DNA repetitive areas, including minisatellites, microsatellites, and the other smaller tandem repeats that we will discuss, gives an immediate indication of the repeat length T. If the DFT window size N is sufficiently large to capture the fundamental frequency $k = N/T$, then all the harmonics will appear as equally spaced horizontal lines at the integer multiples of N/T up to (and including if present) the "maximum" frequency $N/2$. Therefore, the number L of horizontal lines that appear in the spectrogram (without counting the omnipresent DC frequency) will be the integer part of half the repeat length T. Conversely, the repeat length can be deduced by inspection of the spectrogram as $2L$ if L is even, or $2L + 1$ if L is odd. The color of each harmonic shows the contribution from the different bases.

Intergenic tandem repeats are interesting because of their mutagenic properties. It is known that there are large numbers of intergenic tandem repeats in the form of microsatellites and minisatellites in higher organisms. In *C. elegans*, there are around 38 defined dispersed repeat families, many of which correspond to transposon-like elements. Many transposons have already been defined in *C. elegans*

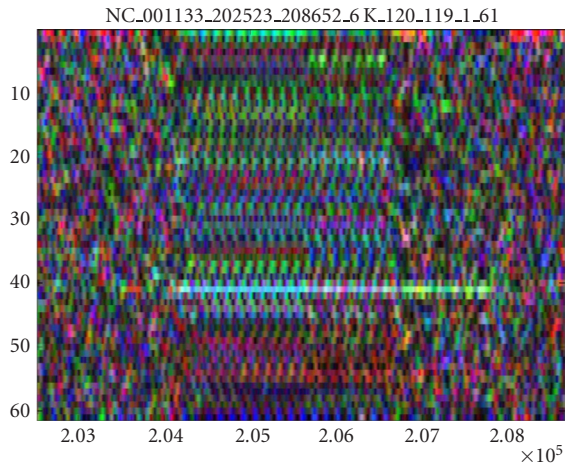


Figure 18: Spectrogram showing the quilt in protein FLO1 corresponding to the flocculin domain.

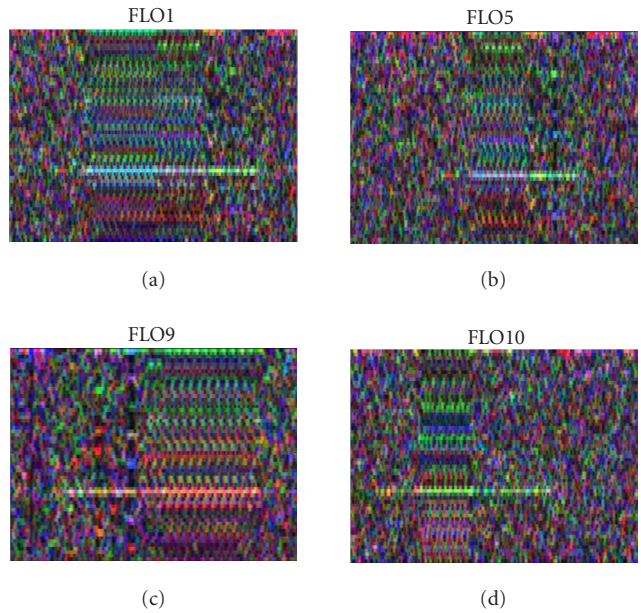


Figure 19: Four spectrograms of FLO genes 1, 5, 9, and 10. Quilts can be seen in all four genes. Close inspection of (a) and (b) shows that (b) is a subsection of (a). FLO9 (c) shows the same coloration as the other three upon reverse complementation.

as mutagenic elements. Many of the dispersed repeat families have been found to be relics of transposon families no longer active. Autosome arms tend to have high recombination rates as compared to the central regions. We found that spectrogram analysis confirms that there are relatively large numbers of repeat patterns in the autosome arms. Some of these repeat clusters were also found in closely related genes. This suggests that these regions may be sites of random mutations and may be rapidly evolving to give rise to new genes and gene families.

3.2. Smaller tandem repeats—quilts, shafts, and bars

After detailed analysis of all the 16 nuclear chromosomes of *S. cerevisiae* (GenBank accession numbers NC_001133-NC_001148) as well as sections of the *C. elegans*, *D. melanogaster*, and human genomes, we identified three basic types of patterns, to which we refer as "quilts," "shafts," and "bars," based on their appearance. All three patterns represent tandem repeats, but the repeat-unit length differs between them. These were not found to be exhaustive but merely illustrative of patterns in the various genomes. Many genes were found to be composites of these patterns. We discovered that quilts, shafts, and bars could be used to predict the homology, structure, and function of proteins. In yeast, most of these patterns were part of the protein-coding regions. However, in the higher organisms, the patterns were also found in the intergenic and intronic regions.

Quilts (Figure 18) are relatively rare patterns in the yeast genome. They appear as beating, repetitive patterns at almost all frequencies over relatively long stretches of DNA. If present in the coding regions of genes, quilts represent protein domains consisting of large tandem repeats. We found quilts representing repeats of up to 45 amino acids (135 bp).

Bars (Figures 20 and 21) and shafts (Figure 22) show strong periodicities uniformly over a stretch of coding DNA. Shafts differ from bars in that they are thin and have few dominant periodicities, causing black areas along most of the other frequencies in the spectrograms. In other words, the basic repeat sequence is smaller in shafts than bars. Bars and quilts with similar appearances, having similar frequency patterns and colors, were found to be homologous as confirmed by BLAST alignment scores, database annotations, and literature.

It should be noted that a quilt appears as a quilt and not as a bar because the DFT window size (typically 120 for viewing proteins) used to create these spectrograms is smaller than the base repeat unit length (135 bp in this case). Although the distinction between quilts and bars is artificial, we found the distinction to be useful since we could differentiate high complexity repeats from lower complexity repeats while still maintaining an appropriate sequence resolution for viewing protein-coding regions.

3.2.1. Quilts—yeast flocculation genes

The quilt observed in Figure 18 is an example of a yeast "flocculation" gene [18]. Yeast flocculation is an asexual, calcium-dependent, and reversible aggregation of cells into flocs. This phenomenon is thought to involve cell surface components. Yeast flocculation is under genetic control, and two dominant flocculation genes have been defined by classical genetics, FLO1 and FLO5. The other relevant FLO genes include FLO9 and FLO10. The functional active domain in these cell surface proteins is made of large tandem repeats up to 45 amino acids known as flocculin repeats. The flocculin region corresponds to the quilted region of the spectrogram. The quilted region was observed in all the FLO genes (Figure 19). The flocculin domain is serine-threonine rich and highly O-glycosylated, adopting a stiff and extended conformation. The efficiency of interaction of the FLO proteins is directly

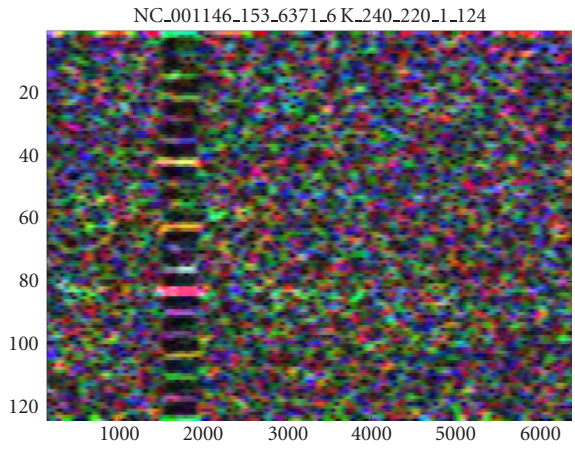


Figure 20: Spectrogram of the YRF1-6 gene. The bar region corresponds to a highly conserved domain in Y′-helicase subtelomeric open reading frames.

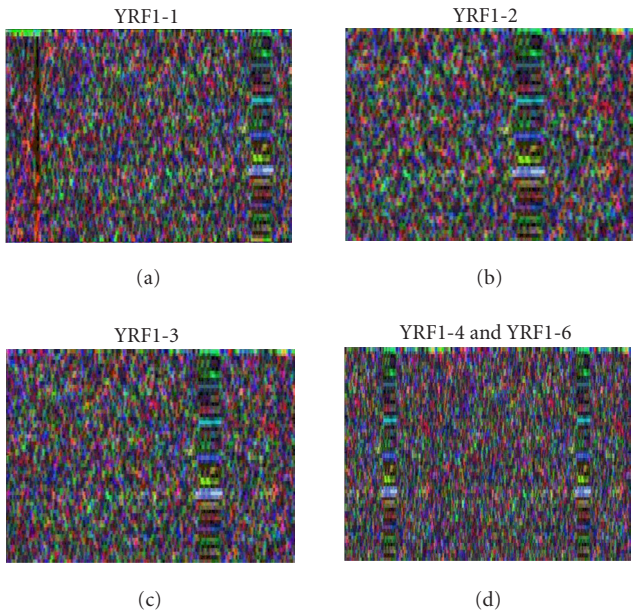


Figure 21: Four spectrograms showing similarity between YRF1 genes 1, 2, 3, 4, and 5. The genes have very similar spectrograms.

dependent on the length of the repeated sequences which are thought to act as spacers to expose a reacting domain at the cell surface. The flocculin repeats that endow the protein with a crucial part of its function are directly visible in the color spectrogram.

Other cell wall proteins whose DNA sequences show quilts are FIT1 (cell wall iron transport) and DAN4 (cell wall mannoprotein). The human MUC2 protein encoded in chromosome 11 of the human genome also shows a large quilt spanning several thousands of nucleotides. This protein is found to have a high BLAST alignment score with FLO1. It is a secreted surface protein that coats the epithelia of intestines, airways, and other mucus membrane containing organs. A common feature is that these proteins have their localization in and around the cellular membrane. Thus, it is possible that the domains represented by quilts cause their proteins to have particular conformations and/or binding sites that function along the cell surface or lead to cell surface localization.

3.2.2. Bars—the Y′-helicases

A large number of bars were found in all genomes, including the yeast genome. We found bars corresponding to protein domains of low complexity tandem repeat units. These repeat units are much simpler, compared to quilts or minisatellites.

The yeast Y′-element is a highly polymorphic repetitive sequence present in the subtelomeric regions of many yeast telomeres [19]. It has been reported that survivors arising from yeast mutants deficient in telomerase compensate for telomere loss by the amplification of Y′-elements. Many of the sequences were found to contain long open-reading frames that potentially encode helicase. Thus, the repetitive patterns in these genes might have a dual role to play. They could function similar to telomeric repeats in extending the life of a cell line. They could also function as important protein domains that are responsible for the helicase function. The Y′-elements contain some highly conserved domains of repeats. One which such domain identified as Pfam-B_59 in the PFAM database shows a unique bar (Figure 20) compared to the other Y′-elements. The helicases that showed bars are

Chromosome 4: YRF1-1/YDR543W
(Bar: 1530000–1530500 bp)
Chromosome 5: YRF1-2/YER190W
(Bar: 574900–575400 bp)
Chromosome 7: YRF1-3/YGR296W
(Bar: 1089000–1089400 bp)
Chromosome 12: YRF1-4/YLR466W
(Bar: 1069500–1070000 bp)
Chromosome 12: YRF1-5/YLR467W
(Bar: 1076250–1076750 bp)
Chromosome 14: YRF1-6/YNL339C
(Bar: 1600–2000 bp)
Chromosome 16: YRF1-7/YPL283C
(Bar: 1500–2000 bp).

Figure 21 shows helicases YRF1-1, YRF1-2, YRF1-3, YRF1-4, and YRF1-5. Part of the conserved domain is seen as a bar.

A large number of other subtelomeric genes show exactly the same bars with the same frequency and color characteristics. The genes are annotated as hypothetical ORFs with unknown functions. The proteins produced from these genes are found to have the same conserved Pfam-B_59 domain. The bar patterns found through spectrogram visualization support the hypothesis that the ORFs have similar functions to the Y′-elements.

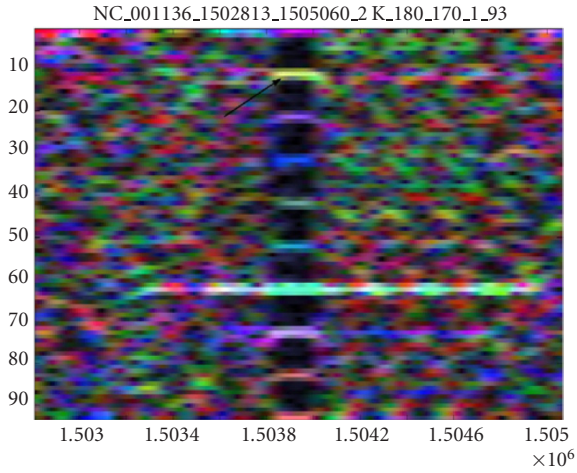


Figure 22: Spectrogram showing shaft in FIT1 gene. The arrow highlights period 18, showing an intensity corresponding to a repeat of 6 amino acids.

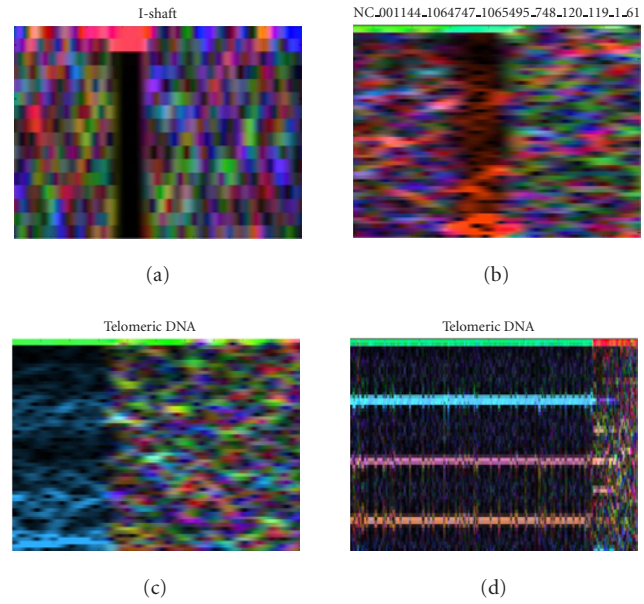


Figure 23: Four spectrograms showing very simple regions. (a) and (b) correspond to simple (1 and 3 bp) repeats in intergenic regions, while (c) and (d) show subtelomeric DNA found at the end of chromosomes.

A number of yeast cell wall glycoproteins such as PIR1, PIR3, HSP150, and TIR1 are characterized by the presence of tandem repeats of a region of 18 to 19 residues. The core region is highly conserved and has a consensus pattern of "SQ [IV] [STGNH] DSQ [LIV] Q [AIV] [STA]." The genomic DNA sequences of these proteins show prominent and characteristic bars whose frequency pattern represents the dominant periodicities. These bars are visually distinct in color and frequency pattern from the Y′-elements.

Some bars show the structural significance of protein in the cell. In yeast, the protein HKR1 coded on chromosome 4 is a cell surface protein that may regulate cell wall beta-glucan synthesis. A region of the gene shows strong bars at a number of relevant frequencies reflecting corresponding periodicities in the protein as well as the DNA sequences. The domain in the protein sequence is made up of 12 repeats of a 28 amino acid sequence, namely, "S [AV] [P] VAVSSTYTSSPSAPAAISSTYTSSP." It was predicted to have a beta-helix supersecondary structure with a high score by the Betawrap algorithm. The gene YIL169C in *S. cerevisiae* shows strong bars that correspond to a serine-rich domain in the protein. This domain extends through amino acids 92–154 and is identified as a potential T-SNARE coiled-coil domain.

3.2.3. Shafts and their structural significance

The shaft shown in Figure 22 is part of the FIT1 gene. It corresponds to a domain of repeats of 6 amino acids, namely, "SSAVET." The shaft shows a bright band at frequency 11, marked by an arrow. The remaining bars are all harmonics of this fundamental periodicity. As the DFT window size was 180 for this spectrogram, a frequency of 11 corresponds to a periodicity of 18 in the DNA sequence and a periodicity of 6 in the protein sequence. This protein domain is predicted with high probability as a large alpha helix by GCG-Pepstruct. Spectrogram analysis of genes CYC8 and GAL11 also show shafts with a prominent periodicity of 6 nucleotides. This translates to a periodicity of 2 amino acids in the protein. Rightly so, they represent QA repeats that form large alpha helices in both proteins.

Many shafts also represent low complexity, high flexibility regions made of GOR turns in the respective proteins. Gene YLR114C has a DENN (differentially expressed in neoplastic versus normal cells) domain. Part of this domain is a high flexibility region of *D* repeats. This region corresponds to a shaft.

Finally, found in the yeast genome were the simplest patterns possible. Some examples of very simple patterns are shown in Figure 23. Very simple repeats of a single to a few nucleotides create simple spectrograms with bright and dark regions. The simplest pattern possible is a dark vertical bar corresponding to a constant nucleotide sequence (e.g., TTTTTTT . . .). These patterns may correspond to subtelomeric DNA or to simple structures in protein-coding regions. Very simple patterns are useful because they serve as visual markers when navigating the genome.

3.2.4. An unannotated pattern

We observed a bar (with many strong periodicities) and a shaft in the region of 12500–13000 nucleotides of *S. cerevisiae* chromosome 1 (Figure 24). Except for this one pattern, every occurrence of quilts, bars, and shafts in the yeast genome was found to correspond to a gene. This region also shows a dominant 3 bp periodicity (the codon frequency). It is sandwiched between 2 genes (12047-12427 and 13364-13744). We found this region to be unannotated in the GenBank and other major databases. Based on these observations, we believe that the region might correspond to a missed gene or pseudogene.

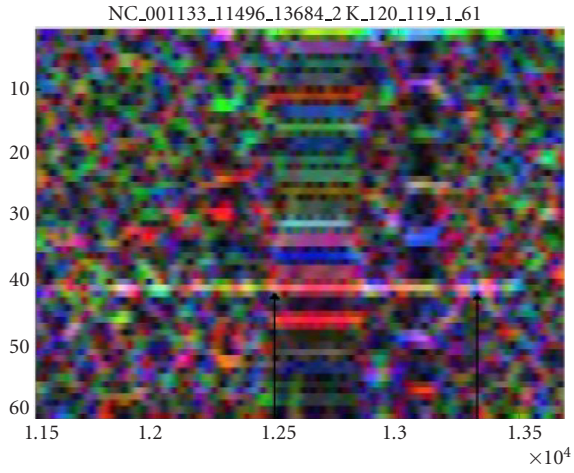


Figure 24: Spectrogram showing an unannotated pattern believed to correspond to a gene or pseudogene. The left arrow marks the end of a predicted gene. The right arrow marks the beginning of another predicted gene.

4. DISCUSSION AND CONCLUSIONS

We employed the short time Fourier transform (STFT) to create color spectrograms of the genomes of various organisms after developing a software tool allowing for easy visual navigation of the genomes via the spectrogram. Spectrograms were created for many different organisms of varying complexity, and we believe that the method can effectively identify any unusual patterns in any genome. Various structures and periodicities were found along all lengths of the chromosome, from a single gene to an entire chromosome. Important periodicities ranged from 0 to 300. We learned that there were no complex patterns in the phage genome and the number of complex patterns increased in frequency with the complexity of the organism. The higher organisms also showed more complex patterns per gene.

Periodicities from 0 to 300 were located and highlighted. We found periodicities relevant to the structure of DNA as well as periodicities involved in protein coding. Periodicities relevant to DNA structure included those concerning telomere structure, protein coding in DNA, DNA helical folding, DNA nucleosome binding, and DNA nucleosome superstructure. One of the characteristics of spectrogram color was that it correlated to Giemsa staining in human chromosomes, thus providing visual information regarding relative nucleotide content, including GC content. Minisatellites were easily visualized as well as the complexity of their constituent repeat pattern.

Patterns of quilts, bars, and shafts were also found on the sequence scale of individual genes. Although bars and shafts were restricted to protein-coding regions in the yeast genome, the same was not true for the higher organisms. In *C. elegans* and humans, some patterns extended into the introns of genes and many were also present in intergenic regions. Patterns were useful in associating homology between various proteins. They were also found to have biological significance, particularly in describing the structure of cell surface proteins. Many classes of cell surface proteins are known and within these classes, there also exist many variants. Cell surface proteins are involved in pathology, pharmacology, and cell signaling. Spectrogram analysis seems particularly well suited for the analysis of this important class of proteins.

A significant challenge in bioinformatics is finding sensible ways to manage the quantity and complexity of information in the genome. Spectrogram analysis of genomes exposes both sequence and frequency information on many scales of magnitude and therefore provides an almost unique visualization of DNA on any magnitude scale. We believe that, based on visual similarity of pattern type such as prominent periodicities and color, this method of frequency analysis is useful as a visualization tool. We found the tool to be particularly useful when used along with public databases and genome browsers. Spectrogram visualization gives a region of DNA a unique visual signature that is useful in quickly recognizing an area of interest. Though spectrograms are much more dynamic, they provide a road map similar to cytological maps used with the fruit fly. Further, this unique visual signature can also be used as a heuristic method of classifying domains in DNA protein-coding regions. Finally, the spectrogram gives insight regarding the physical structure of DNA in which a sequence of interest is embedded. Thus, DNA color spectrograms place sequences of interest in a much-needed larger context.

In summary, we used DNA color spectrograms to find biologically relevant patterns in the genomes of various organisms, some of which relate to DNA structure or protein coding. Similar patterns in different parts of various genomes were found to have similar functions. Various patterns included strong genome-wide periodicities and structures such as microsatellites, minisatellites, quilts, bars, and shafts. We believe that spectrogram analysis will be a useful tool in understanding the DNA structure, identifying protein domains, and predicting function and structure, as well as a discovery tool for novel DNA regions of potential biological significance.

ACKNOWLEDGMENT

Appreciation is expressed to Rick Thompson who introduced the terms “quilts,” “shafts,” and “bars,” and to Chris Fidyk who wrote the original software, implementing spectrogram development.

REFERENCES

[1] D. Anastassiou, “Frequency-domain analysis of biomolecular sequences,” *Bioinformatics*, vol. 16, no. 12, pp. 1073–1081, 2000.

[2] D. Anastassiou, “Genomic signal processing,” *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 8–20, 2001.

[3] J. C. Shepherd, “Periodic correlations in DNA sequences and evidence suggesting their evolutionary origin in a comma-less genetic code,” *J. Mol. Evol.*, vol. 17, no. 2, pp. 94–102, 1981.

[4] J. C. Shepherd, “Method to determine the reading frame of a protein from the purine/pyrimidine genome sequence and

its possible evolutionary justification," *Proc. Natl. Acad. Sci. USA*, vol. 78, no. 3, pp. 1596–1600, 1981.
[5] J. C. Shepherd, "From primeval message to present-day gene," *Cold Spring Harb. Symp. Quant. Biol.*, vol. 47, Pt 2, pp. 1099–1108, 1983.
[6] J. W. Fickett, "Recognition of protein coding regions in DNA sequences," *Nucl. Acids. Res.*, vol. 10, pp. 5303–5318, 1982.
[7] D. Rhodes and A. Klug, "Helical periodicity of DNA determined by enzyme digestion," *Nature (London)*, vol. 286, pp. 573–578, August 1980.
[8] G. P. Lomonossoff, P. J. Butler, and A. Klug, "Sequence-dependent variation in the conformation of DNA," *J. Mol. Biol.*, vol. 149, pp. 745–760, July 1981.
[9] A. Klug, L. C. Lutter, and D. Rhodes, "Helical periodicity of DNA on and off the nucleosome as probed by nucleases," *Cold Spring Harb. Symp. Quant. Biol.*, vol. 47, pp. 285–292, 1983.
[10] L. J. Peck and J. C. Wang, "Sequence dependence of the helical repeat of DNA in solution," *Nature*, vol. 292, pp. 375–378, July 1981.
[11] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, Garland Publishing, New York, USA, 4th edition, Chapter 4, 2002.
[12] Y. Niimura and T. Gogobori, "*In silico* chromosome staining: Reconstruction of Giemsa bands from the whole human genome sequence," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 2, pp. 797–802, 2002.
[13] A. Bird, "CpG islands as gene markers in the vertebrate nucleus," *Trends in Genetics*, vol. 3, pp. 342–347, 1987.
[14] S. Subramanian, V. M. Madgula, R. George, et al., "Triplet repeats in human genome: distribution and their association with genes and other genomic regions," *Bioinformatics*, vol. 19, no. 5, pp. 549–552, 2003.
[15] S. Subramanian, R. K. Mishra, and L. Singh, "Genome-wide analysis of Bkm sequences (GATA repeats): predominant association with sex chromosomes and potential role in chromatin organization and function," *Bioinformatics*, vol. 19, no. 6, pp. 681–685, 2003.
[16] K. B. Murray, D. Gorse, and J. Thornton, "Wavelet transforms for the characterization and detection of repeating motifs," *J. Mol. Biol.*, vol. 316, no. 2, pp. 341–363, 2002.
[17] Y. Nakamura, M. Leppert, P. O'Connell, et al., "Variable number of tandem repeat (VNTR) markers for human gene mapping science," *Science*, vol. 235, no. 4796, pp. 1616–1622, 1987.
[18] M. Bony, D. Thines-Sempoux, P. Barre, and B. Blondin, "Localization and cell surface anchoring of the Saccharomyces cerevisiae flocculation protein Flo1p," *Journal of Bacteriology*, vol. 179, no. 15, pp. 4929–4936, 1997.
[19] M. Yamada, N. Hayatsu, A. Matsuura, and F. Ishikawa, "Y′-Help1, a DNA helicase encoded by the yeast subtelomeric Y′ element, is induced in survivors defective for telomerase," *J. Biol. Chem.*, vol. 273, no. 50, pp. 33360–33366, 1998.

David Sussillo received his B.S. degree in computer science from Carnegie Mellon University in 1999 and his M.S. degree in electrical engineering from Columbia University in 2003. He is currently pursuing his Ph.D. degree in the Doctoral Program for Neurobiology and Behavior at Columbia University. His current research interests include signal processing of genomic signals, vision processing in the primary visual cortex, and computer applications in biomedical research.

Anshul Kundaje received his B.S. degree from Veermata Jijabai Technological Institute (VJTI), the University of Mumbai in 2001 and M.S. degree from Columbia University in 2002, both in electrical engineering. Presently, he is pursuing a Ph.D. degree in computer science at Columbia University. His research focus is computational biology, specifically applying machine learning and signal processing techniques to solving hard biological problems. His prime interest is in reverse engineering of genetic and protein networks using multiple sources of biological data such as mRNA expression, time-series, sequence, and protein data.

Dimitris Anastassiou is a Professor and Director of Columbia's Genomic Information Systems Laboratory at Columbia University. He received the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 1979. From 1979 to 1983, he was a research staff member at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. Since 1983, he has been with the Department of Electrical Engineering, Columbia University. He is an IEEE Fellow, the recipient of an IBM Outstanding Innovation Award, a National Science Foundation Presidential Young Investigator Award, and a Columbia University Great Teacher Award. His previous research interests have been in the area of digital signal processing and information theory with emphasis on the digital representation of multimedia signals, with contributions to the international digital television coding standard, MPEG-2. He is the founder and previous Director of Columbia University's Image and Advanced Television Laboratory. His research is now exclusively focused on applying his expertise in engineering to the emerging field of computational biology.

EURASIP Journal on Applied Signal Processing 2004:1, 43–52

Multicriteria Gene Screening for Analysis of Differential Expression with DNA Microarrays

Alfred O. Hero

Departments of Electrical Engineering and Computer Science, Biomedical Engineering, and Statistics, University of Michigan, Ann Arbor, MI 48109, USA
Email: hero@eecs.umich.edu

Gilles Fleury

Service des Mesures, Ecole Supérieure d'Electricité, 91192 Gif-sur-Yvette, France
Email: fleury@supelec.fr

Alan J. Mears

Departments of Ophthalmology and Visual Sciences, and Human Genetics, University of Michigan Medical School, Ann Arbor, MI 48109, USA

University of Ottawa Eye Institute, Ottawa Health Research Institute, Ottawa, ON Canada, K1H 8L6
Email: amears@ohri.ca

Anand Swaroop

Departments of Ophthalmology and Visual Sciences, and Human Genetics, University of Michigan Medical School, Ann Arbor, MI 48109, USA
Email: swaroop@med.umich.edu

Received 10 May 2003; Revised 30 August 2003

This paper introduces a statistical methodology for the identification of differentially expressed genes in DNA microarray experiments based on multiple criteria. These criteria are false discovery rate (FDR), variance-normalized differential expression levels (paired t statistics), and minimum acceptable difference (MAD). The methodology also provides a set of simultaneous FDR confidence intervals on the true expression differences. The analysis can be implemented as a two-stage algorithm in which there is an initial screen that controls only FDR, which is then followed by a second screen which controls both FDR and MAD. It can also be implemented by computing and thresholding the set of FDR P values for each gene that satisfies the MAD criterion. We illustrate the procedure to identify differentially expressed genes from a wild type versus knockout comparison of microarray data.

Keywords and phrases: bioinformatics, gene filtering, gene profiling multiple comparisons, familywise error rates.

1. INTRODUCTION

Since Watson and Crick discovered DNA more than fifty years ago, the field of genomics has progressed from a speculative science to one of the most thriving areas of current research and development [1]. After successful completion (99%) of the Human Genome project [2], attention is turning to "functional genomics" and "proteomics," thanks principally to remarkable advances in computations and technology. These disciplines encompass the greater challenge of understanding the complex functional behavior and interaction of genes and their encoded proteins at the cellular level. This task has been significantly aided by the advent of DNA microarray technology and associated algorithms that enable researchers to filter through daunting amounts of data and genetic information. In this paper, we describe a new approach to extracting a subset of differentially expressed genes from DNA microarray data.

A DNA microarray consists of a large number of DNA probe sequences that are put at defined positions on a solid support such as a glass slide or a silicon wafer [3, 4]. After hybridization of a fluorescently labelled sample (gene transcripts) to DNA microarrays, the abundance of each probe present (called probe response) in the sample can be estimated from the measured levels of hybridization (i.e., the intensity of fluorescent signal). Two main types of DNA microarrays are in wide use for gene expression profiling: Affymetrix GeneChips [5], which are generated by photolithography; and spotted cDNA (or oligonucleotide) arrays on glass slides [6].

DNA microarrays enable biologists to study global gene expression profiles in tissues of interest over time periods and under specific conditions or treatments. For these cases, a large set of samples, consisting of several biological replicates, are hybridized to a set of microarrays. The objective is to identify subsets of genes whose expression profile over time exhibit salient behavior(s), for example, differ in response to different treatments. A crucial aspect of selecting the genes of interest is the specification of a preference ordering for ranking the probe responses. Many gene selection and ranking methods are based on testing fitness criteria such as the eigenvalue spread in a principal components analysis (PCA) of all pairs of gene expression profiles, the ratio of between-population-variation to within-population-variation, or the cross correlation between profiles [7, 8, 9].

These methods have deficiencies which have impeded their use for practical experiments. First, is the need for improved relevance of the fitness criterion to the scientific objectives of the experiment. It is often difficult for an experimenter to choose quantitative criteria that characterize the aspects of a gene expression profile of interest. Second, is the need for simultaneous control of the biological significance (minimum acceptable difference (MAD)) and the statistical significance (false discovery rate (FDR)) of differential responses discovered in the selected gene probes. A probe response difference which is too small is not of much use to the experimenter even if the difference is statistically significant. This is because the microarray experiment is usually only the first step in gene discovery; each microarray probe difference that is discovered must be validated by painstaking-followup analysis that may have limited sensitivity to small differences. Third, is the need for tight confidence intervals (CIs) on these differences. The size of a CI provides useful information on the statistical precision of an estimate of differential response.

The method we present in this paper adopts a statistical multicriteria framework for gene microarray analysis with MAD constraints on differential expression. The framework allows the experimenter to adopt multiple fitness criteria, explicitly incorporate control on biological significance in addition to statistical significance, and generate confidence intervals on discovered gene expression differences. Our method is strongly influenced by the FDR-adjusted confidence interval (FDR-CI) approach recently introduced by Benjamini and Yekutieli [10]. We illustrate our methods for a differential expression experiment designed to probe the genetic basis of retinal development. This experiment involves two populations, wild type and knockout, and the objective is to find genes that exhibit biologically and statistically significant differences between these populations. The purpose of this article is to illustrate methodology and not to report scientific findings, which will be reported elsewhere.

It is worthwhile to compare the framework developed in this paper to related work. Liu and Iba have proposed an interesting multicriteria evolutionary approach to gene selection and classification in gene microarray experiments [11]. Similarly, Fleury and Hero have proposed Pareto optimality for selecting subsets of genes using a combination of bootstrap resampling and Bayes decision theory [12, 13, 14]. Single stage [15] and multistage [16, 17, 18] screening methods which control familywise error rate (FWER) or FDR have been proposed by several authors for similar problems to ours. However, none of the above approaches account for a MAD constraint or provide CIs on the differential expression levels of the discovered genes. In contrast, our approach accounts for both FDR and MAD constraints and generates such confidence intervals using the FDR-CI framework [10]. Furthermore, we specify an algorithm for computing FDR P values for all genes at any prescribed MAD level.

Table 1: The knockout versus wild-type experiment is equivalent to a two-way layout of treatment (W or K) and time (t = Pn2, Pn10, M2).

Gene g	Pn2	Pn10	M2
W	4 samples	4 samples	4 samples
K	4 samples	4 samples	4 samples

The outline of the paper is as follows. In Section 2, we give a general description of the type of differential gene microarray experiment that will be illustrated in Section 4. In Section 3, we describe the proposed two-stage multicriteria approach. Finally, in Section 4, we illustrate these techniques for experimental data.

2. DIFFERENTIAL EXPRESSION PROFILE EXPERIMENTS

This type of experiment is very common in genetics research [19, 20] and involves comparing gene expression profiles of a set of G genes expressed in two or more populations. The data from this experiment fall into the category of a two-way layout [21], where each cell in the layout corresponds to a set of replicates of samples from one of the two populations (row) and one of T-time points (column) (see Table 1).

Any gene whose temporal profile differs from wild-type to knockout populations is called "differentially expressed" in the experiment. One variant of this experiment is called the wild-type versus knockout experiment. In such an experiment, one has a control population (wild type) of subjects and a treated population (knockout) of subjects whose DNA has been altered in some way. Each population is comprised of T different age groups arranged in T subpopulations. M independent samples are taken from each subpopulation and are hybridized to a different microarray, yielding G pairs of expression profiles (see Figure 1 for profiles of the gene having probe set number 101996_*at*). This generates a total of $2MT$ microarrays. It is common to express the differential response between wild-type and knockout responses in terms of *foldchange* expressed as the ratio of these responses. For example, a foldchange of 2.0, or 1.0 in log base 2 at a given time corresponds to a wild-type response which is twice as large as the knockout response. We denote by $\{\mu_t(g)\}_{t=1}^{T}$ and $\{\eta_t(g)\}_{t=1}^{T}$ the true log wild-type and log knockout expression profiles, respectively, expressed as log base 2 of the true hybridization abundances.

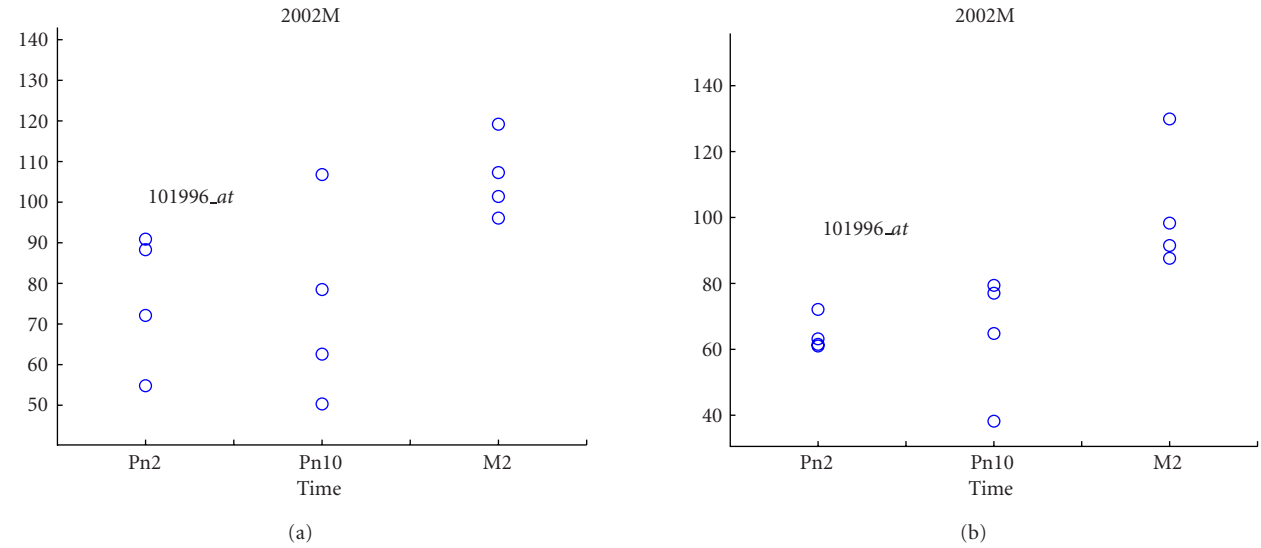


Figure 1: Responses for a particular gene (probe set number 101996_*at*) in (a) knockout mouse versus (b) wild-type mouse for the differential expression study discussed in Section 4. There are three-time points (labeled Pn2, Pn10, and M2) and at each time point, there are four replicates. The y-axis denotes log base 2 hybridization level extracted by RMA from Affymetrix GeneChips.

Figure 2 illustrates the three-dimensional multicriteria space of mean differential responses $\{\mu_t(g) - \eta_t(g)\}_{t=1}^{3}$ for the three-time point experiment described in Section 4. A "MAD box" which defines unacceptably small (inside box) versus acceptably large (outside box) differential responses, and a scatter of a small subset of all the sample mean differential responses (dots) from the experiment are also indicated. Our objective is to discover which genes are likely to have a "positive differential response" falling outside of the box in Figure 2. A very commonly used method is to simply apply a threshold to the sample means to detect those who fall outside of the box in Figure 2 as positive responses. However, as will be shown, this method does not account for statistical sampling uncertainty and can lead to many false positives.

The objective can be stated mathematically as follows: find a set of gene probes which satisfy the MAD constraint: $|\mu_t(g) - \eta_t(g)| > \text{fcmin}$ for at least one $t \in \{1, \ldots, T\}$. Here, the MAD constraint is quantified by the user-specified minimum magnitude foldchange fcmin (expressed in log base 2). Thus, we need to simultaneously test the G pairs of the two-sided hypotheses

$$\begin{aligned} H_0(g) &: |\mu_1(g) - \eta_1(g)| \\ &\quad \le \text{fcmin and, } \cdots, \text{and } |\mu_T(g) - \eta_T(g)| \\ &\quad \le \text{fcmin}, \\ H_1(g) &: |\mu_1(g) - \eta_1(g)| \\ &\quad > \text{fcmin or, } \cdots, \text{or } |\mu_T(g) - \eta_T(g)| \\ &\quad > \text{fcmin}, \end{aligned} \tag{1}$$

where $g = 1, \ldots, G$. Of course, when we must decide between $H_0(g)$ and $H_1(g)$ based on a random sample, there will generally be decision errors in the form of false positives (decide $H_1(g)$ when $H_0(g)$ is true) and false negatives (decide $H_0(g)$ when $H_1(g)$ is true). For any test, the experimenter needs to be able to control both its statistical and biological level of significance. The *statistical level of significance* of the test is specified by the false positive rate. In contrast, the *biological level of significance* of the test is specified by fcmin.

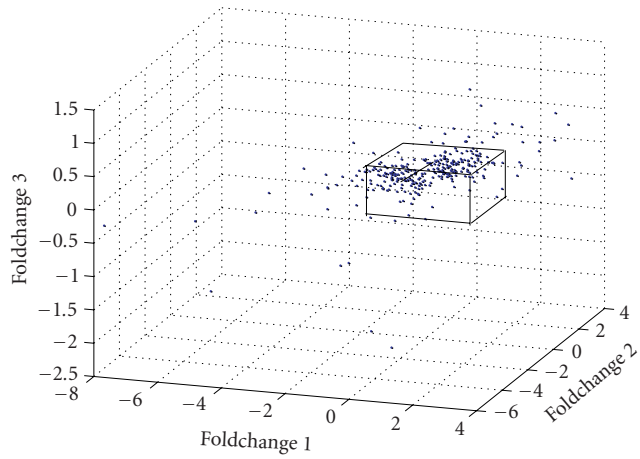


Figure 2: Three-dimensional multicriteria space for knockout and wild-type profiles over three-time points shown in Figure 1. The three criteria are the differential probe responses at each time point. A scatter plot of sample means of the differential responses along with a box of edge length 2fcmin distinguishing biologically significant responses (outside box) from biologically insignificant responses (inside box) is shown.

There are three aspects to the hypothesis-testing problem (1) which make it nonstandard:

(i) standard tests on differences in means, such as the paired t test, treat any nonzero difference as significant,

whereas (1) specifies that only differences exceeding the specified MAD level of fcmin are significant;

(ii) a positive response ($H_1(g)$) is described by multiple criteria, here equal to the T magnitude log response ratios at each point in time;

(iii) the G pairs of hypotheses must be tested simultaneously.

For the case $G = T = 1$, the first aspect can be treated by applying methods for composite hypothesis testing such as generalized likelihood ratio tests, unbiased tests, and CI test procedures [22, 23]. When fcmin $= 0$, (ii) and (iii) can be handled by applying a standard method, like paired t-test, to (1) for each gene probe g, implemented with a multiplicity error-correction factor, for example, Bonferroni, FWDR, or FDR, [24]. However, such a repeated test of significance will result in excessive false positives corresponding to small log response ratios that are biologically insignificant (do not satisfy the MAD constraint) but are statistically significant.

3. MULTICRITERIA GENE SCREENING METHOD

Define $\underline{\xi}(g) = [\xi_1(g), \ldots, \xi_T(g)]$ the true differential response vector associated with gene probe g, where $\xi_t(g) = \mu_t(g) - \eta_t(g)$. Given the DNA microarray data, our objective is to test the G hypotheses (1) involving a total of $P = GT$ unknown parameters $\{\underline{\xi}(g)\}_{g=1}^{G}$.

Any test of (1) must test over multiple criteria $\{\underline{\xi}_t(g)\}_t$ and multiple genes at a given level of biological significance MAD $=$ fcmin and a given level of statistical significance max FDR $= \alpha$. Unless fcmin $= 0$, this is a doubly composite hypothesis-testing problem since the parameter values ξ_t are not specified under H_0 or H_1. Due to the presence of multiple criteria and multiple genes, this problem falls into the area of multiple testing, simultaneous inference, and repeated tests of significance [25, 26]. Two standard measures of statistical significance of a test of (1) are its FWER and its FDR [25]. A mathematically convenient notation for a test of (1) is $\phi(g)$, which is called a *test function*, taking on values 0 or 1 depending on whether the test declares H_0 or H_1 for probe g, respectively. With $\mathcal{G}_0$ denoting the probes not having positive responses, the FWER and FDR of a test ϕ can be mathematically defined as

$$\begin{aligned} \text{FWER}\,(\mathcal{G}_0) &= 1 - E\left[\Pi_{g=1}^{G}(1-\phi(g))\psi_{\mathcal{G}_0}(g)\right], \\ \text{FDR}\,(\mathcal{G}_0) &= E\left[\frac{\sum_{g=1}^{G}\phi(g)\psi_{\mathcal{G}_0}(g)}{\sum_{g=1}^{G}\phi(g)}\right], \end{aligned} \tag{2}$$

where $E[Z]$ denotes statistical expectation of a random variable Z and $\psi_{\mathcal{G}_0}(g)$ is the indicator function of the set $\mathcal{G}_0$. In words, the FWER is the probability that the test of all G pairs of hypotheses (1) yields at least one false positive in the set of declared positive responses. In contrast, the FDR is the average proportion of false positives in the set of declared positive responses. The FDR is dominated by the FWER and is therefore a less stringent measure of significance. Both FWER and FDR have been widely used for gene microarray analysis [16, 17, 24, 27].

It is useful to contrast the FWER and FDR to the per-comparison error rate (PCER). The PCER refers to the false positive error rate incurred in testing a single pair of hypothesis $H_0(g)$ versus $H_1(g)$ for a single gene, say, gene $g = g_o$, and does not account for multiplicity of the hypotheses (1). The PCER is the probability that random sampling errors would have caused g_o to be erroneously selected, generating a false positive, based on observing microarray responses for gene g_o only. If an experimenter were only interested in deciding on the biological significance of a single gene g_o, based only on observing probes for that gene, then reporting PCER(g_o) would be sufficient for another biologist to assess the statistical significance of the experimenter's statement that g_o exhibits a positive response. In contrast to the PCER, FWER and FDR communicate statistical significance of an experimenter's finding of biological significance after observing all gene responses. The FWER is the probability that there are any false positives among the set of genes selected. On the other hand, the FDR refers to the expected proportion of false positives among the selected genes. The FDR is a less stringent criterion than the FWER [25, 27, 28].

The FWER can be upper bounded as a function of $\{\text{PCER}(g)\}_{g=1}^{G}$ using Bonferroni-type methods [26] or it can be computed empirically from the sample by resampling methods [29]. The FDR can be computed by applying the step-down procedure of Benjamini and Hochberg [25] to the list of PCER P values over all genes. For a given g, the PCER P value, denoted $p(g)$, of a test ϕ is a function of the microarray measurements and is defined as the minimum value of PCER for which $H_0(g)$ would be falsely rejected by the test. The set of gene responses which pass the test ϕ at a specified FDR can be simply determined after ordering the genes indices according to increasing PCER P value $p(g_{(1)}) \leq \cdots \leq p(g_{(G)})$. Specifically, for a fixed value $\alpha \in [0, 1]$ of maximum acceptable FDR, the FDR-constrained test will declare the following set $\mathcal{G}_1$ of genes as positive responses [28]:

$$\begin{aligned} \mathcal{G}_1 &= \{g_{(1)}, \ldots, g_{(K)}\}, \\ K &= \max\left\{k : p(g_{(k)}) \leq \frac{k\alpha}{G\nu}\right\}. \end{aligned} \tag{3}$$

In this expression, $\nu = 1$ if the decisions $\phi(g)$ can be assumed statistically independent over $g = 1, \ldots, G$, while $\nu = 1/\sum_{k=1}^{G} k^{-1}$ without the independence assumption.

A test which controls a maximum level α of acceptable FDR is said to be an FDR test of level α. We propose a test ϕ of (1) at FDR level α and MAD level fcmin based on intersecting simultaneous CIs on the T differences $\underline{\xi}(g)$ with the unacceptable difference region $[-\text{fcmin}, \text{fcmin}]$. We will specify a two-stage direct implementation and a single-stage inverse implementation in the following subsections. First, however, we recall some facts about simultaneous CIs.

Let θ be an unknown parameter, for example, a gene's foldchange $\xi_1(g)$ at time $t = 1$. A PCER $(1-\alpha)\times 100\%$ CI on θ is an interval $I(\alpha) = [a, b]$ with random data-dependent endpoints that covers the true θ value, say θ_o, with probability at least $1 - \alpha$:

$$P(a \le \theta_o \le b \mid \theta = \theta_o) \ge 1 - \alpha. \quad (4)$$

There is always a trade-off between confidence level $1-\alpha$ and precision (CI length) since the length $b-a$ of $I(\alpha)$ generally increases as α decreases. Let $\mathcal{A}$ be any subset of $\mathbb{R}$. A PCER CI on θ can be converted to a PCER level-α test of the hypotheses $H_0(g) : \theta \in \mathcal{A}$ versus $H_1(g) : \theta \notin \mathcal{A}$ by the simple procedure: "reject H_0 if the $(1-\alpha)\times 100\%$ CI on θ does not intersect $\mathcal{A}$" [22].

Multiple parameters, $\theta_1, \ldots, \theta_P$, can be simultaneously covered by FWER $(1-\alpha)\times 100\%$ CIs $\{I^p(1-(1-\alpha)^{1/P})\}_{p=1}^{P}$, where $I^p(\alpha)$ is a PCER $(1-\alpha)\times 100\%$ CI on θ_p. Under the assumption that each of the P PCER CIs are statistically independent, the FWER intervals cover all the parameters with probability at least $1-\alpha$ [26]. A less stringent set of CIs $\{I^p(\alpha/P)\}_{p=1}^{P}$, which can be applied to dependent sets of PCER CIs, is guaranteed to cover at least $(1-\alpha)P$ of the unknown parameters [26, 30]. When the number of P of parameters is random, as occurs when the number of parameters results from some initial screening, the above methods cannot be applied. It was for this situation that the FDR-CI approach was developed [10]. If P is the result of initial screening at an FDR level α of Q parameters having PCER-CIs $\{I^p(\alpha)\}_{p=1}^{Q}$, then the FDR-CIs on the P parameters are defined as $\{I^p(P\alpha/Q)\}_{p=1}^{P}$. The FDR-CIs are guaranteed to cover at least $(1-\alpha)\times 100\%$ of the P unknown parameters.

Below, we give two equivalent FDR-CI procedures for screening differentially expressed genes with FDR and MAD constraints.

3.1. Direct two-stage screening procedure

Stage 1. Gene screening at MAD level 0 extracts a set of G_1 genes $\mathcal{G}_1$ by testing (1) under the relaxed MAD constraint fcmin $= 0$ using an FDR level-α test via the step-down procedure (3).

Stage 2. Gene screening at MAD level fcmin > 0 extracts a set $\mathcal{G}_2$ of positive genes from those in $\mathcal{G}_1$ as follows. For each gene $g \in \mathcal{G}_1$, construct T simultaneous CIs, denoted as $\{I_t^g(\alpha)\}_{t=1}^{T}$, of FWER level $(1-\alpha)\times 100\%$ on the true fold-changes $\{\mu_t(g)-\eta_t(g)\}_{t=1}$. Convert these into $(1-\alpha)\times 100\%$ FDR-CIs by the method of Benjamini and Yekutieli [10]: $I_t^g(\alpha) \to I_t^g(G_1\alpha/G)$, $t = 1, \ldots, T$, $g = 1, \ldots, G$. Finally, define the set of indices $\mathcal{G}_2$ of gene profiles having at least one-time point, where the FDR-CI does not intersect $[-\text{fcmin}, \text{fcmin}]$:

$$\mathcal{G}_2 = \{g \in \mathcal{G}_1 : (\cup_{t=1,2,3} I_t^g(G_1\alpha/G) \cap [-\text{fcmin}, \text{fcmin}]) = \varnothing\}, \quad (5)$$

where $\varnothing$ denotes the empty set. It follows from [10, Section 3.1] that the set $\mathcal{G}_2$ has FDR less than or equal to α at MAD level fcmin.

3.2. Inverse screening procedure: FDR P values

In many practical situations, the experimenter may not be comfortable in specifying a MAD or FDR criterion in advance. In these situations, it is more useful to solve the following "inverse problem:" what is the most stringent pair of criteria (α, fcmin) that would lead to including a particular gene among the positives $\mathcal{G}_2$? For fixed fcmin, the most stringent (minimum) value α for which a gene would fall into $\mathcal{G}_2$ is called the FDR P value. The FDR P value for a gene g_o can be computed by (1) computing the PCER P value sequence $\{p(g)\}_{g=1}^{G}$; (2) arranging the PCER P value sequence in an increasing order $p(g_{(1)}) \le \cdots \le p(g_{(G)})$; (3) finding the minimum value $\alpha = \alpha(g_o)$ for which at least one of the PCER CIs $\{I_t^{g_o}(\alpha)\}_{t=1}^{T}$ does not intersect $[-\text{fcmin}, \text{fcmin}]$; and (4) computing the integer index

$$N(\alpha(g_o)) = \sum_{k=1}^{G} I\left(p(g_{(k)})\frac{k}{G} \le 1 - (1-\alpha(g_o))^T\right), \quad (6)$$

where $I(A) = 1$ if statement A is true and $I(A) = 0$ otherwise; the FDR P value of g_o is then simply $p(g_i)$, where $i = N(\alpha(g_o))$. Repeating this as g_o ranges over $1, \ldots, G$ gives a sequence of FDR P values at MAD level fcmin that can be thresholded to determine the set of positive genes $\mathcal{G}_2$ at any desired FDR level of significance.

4. APPLICATION TO A WILD-TYPE VERSUS KNOCKOUT EXPERIMENT

These experiments were performed to investigate the role of a specific retinal transcription factor Nrl [31] in the development of mouse retina. The retinal samples were taken from four pairs ("biological replicates") of wild-type and knockout (Nrl deficient) mice [32] at three different time points: postnatal day 2 (Pn2), postnatal day 10 (Pn10), and 2 months of age (M2). The samples were then hybridized to a total of twenty-four MGU74Av2 Affymetrix GeneChips. The log base 2 probe responses were extracted from Affymetrix GeneChips using the robust microarray analysis (RMA) package [33]. We denote the measured wild-type and knockout responses by $W_{t,m}(g)$ and $K_{t,m}(g)$, where $m = 1, \ldots, M$, $t = 1, \ldots, T$, and $g = 1, \ldots, G$ are microarray replicate, time, and gene probe location on the microarray, respectively. For this experiment, $G = 12421$, $M = 4$, and $T = 3$. To construct CIs on foldchanges, we define the vector of paired t-test statistics:

$$\underline{\hat{\xi}}(g) = \left[\frac{|\overline{W_1}(g) - \overline{K_1}(g)|}{s_1(g)/\sqrt{M/2}}, \frac{|\overline{W_2}(g) - \overline{K_2}(g)|}{s_2(g)/\sqrt{M/2}}, \frac{|\overline{W_3}(g) - \overline{K_3}(g)|}{s_3(g)/\sqrt{M/2}}\right], \quad (7)$$

where $g = 1, \ldots, G$. Here, $\overline{W_t}(g) = M^{-1}\sum_{m=1}^{M} W_{t,m}(g)$ and $\overline{K_t}(g) = M^{-1}\sum_{m=1}^{M} K_{t,m}(g)$ denote the sample mean of the M replicates at time t for wild-type and knockout treatments, respectively, and

$$s_t^2(g) = (2(M-1))^{-1}\left(\sum_{m=1}^{M}(W_{t,m}(g) - \overline{W_t}(g))^2 + \sum_{m=1}^{M}(K_{t,m}(g) - \overline{K_t}(g))^2\right) \quad (8)$$

denotes the pooled sample variance at time t.

Table 2: Two stage FDR-CI algorithm for screening genes from the knockout versus wild-type experiment.

Stage 1	Compute and sort PCER P values according to (9) Select gene indices $\mathcal{G}_1$ according to (3)
Stage 2	Construct simultaneous PCER CIs using (10) Select gene indices $\mathcal{G}_2$ according to (5)

For Stage 1 of the screening procedure, we consider the simple and standard (see [26]) simultaneous test of (1) at MAD level fcmin $= 0$: "decide $H_1(g)$ if $\max_{t=1,2,3}(|\overline{W_t}(g) - \overline{K_t}(g)|/s_t(g)/\sqrt{M/2}) >$ fcmin." Under the large M approximation that the paired t test statistic has a Student t distribution [34], and assuming time independence of cells in the two-way layout of Table 1, we can easily compute both the PCER P value for this test:

$$p(g) = 1 - \left[2\mathcal{T}_{2(M-1)}(\hat{\xi}(g)) - 1\right]^3, \tag{9}$$

and simultaneous $(1-\alpha)\times 100\%$ CIs, $I_1^g(\alpha)$, $I_2^g(\alpha)$, $I_3^g(\alpha)$, for the temporal foldchanges $\{\mu_t(g) - \eta_t(g)\}_{t=1,2,3}$ of gene g:

$$\begin{aligned} &\overline{W_t}(g) - \overline{K_t}(g) - \frac{s_t(g)}{\sqrt{M/2}\mathcal{T}^{-1}_{2(M-1)}}\left(1-\frac{\alpha}{2}\right) \\ &\quad\le \mu_t(g) - \eta_t(g) \\ &\quad\le \overline{W_t}(g) - \overline{K_t}(g) + \frac{s_t(g)}{\sqrt{M/2}\mathcal{T}^{-1}_{2(M-1)}}\left(1-\frac{\alpha}{2}\right), \end{aligned} \tag{10}$$

$t = 1, 2, 3$. In the above inequality, $\mathcal{T}_\nu : \mathbb{R} \mapsto [0,1]$ denotes the Student t cumulative distribution function with ν degrees of freedom and $\mathcal{T}_\nu^{-1}$ denotes its functional inverse, that is, the Student t quantile function.

With the above expressions, we can find the set $\mathcal{G}_1$ of gene indices which pass Stage 1 FDR screening by substituting the sorted PCER P values (9) into the step-down algorithm (3). Stage 2 of screening selects gene indices according to the FDR-CIs from (5). This direct two-stage screening stage procedure is summarized in Table 2. Alternatively, the inverse procedure of Section 3.2 can be implemented using (9) and the explicit expression for the $\alpha(g)$ sequence

$$\alpha(g) = 2\left[1 - \mathcal{T}_{2(M-1)}\left(\frac{\max|\overline{W_t}(g) - \overline{K_t}(g)| - \text{fcmin}}{s_t(g)/\sqrt{M/2}}\right)\right], \tag{11}$$

where $g = 1, \ldots, G$.

4.1. Experimental results

Figures 3 and 4 illustrate the direct and inverse implementations of the FDR-CI screening procedure. In Figure 3, the direct screening procedure is constrained by MAD and FDR criteria fcmin $= 2.0$ and $\alpha = 0.2$, respectively. As there are ($T = 3$)-time points and $G = 12\,421$ genes, there are $GT = 37\,263$ parameters for which FDR-CIs are constructed. A gene passes the screening if at least one of the three time instants has an FDR-CI that does not intersect the interval $[-\text{fcmin}, \text{fcmin}]$. The test is implemented by defining two rank orderings of the FDR-CIs of the genes according to (1) the FDR-CI with minimum upper boundary over the three time points; and (2) the FDR-CI with maximum lower boundary over the time points. Figures 3a and 3b show relevant segments of these two ordered sequences of CIs. Screening all genes with maximum lower endpoints > fcmin and minimum upper endpoints < −fcmin generates the set of declared positive genes $\mathcal{G}_2$.

Figure 4 illustrates the inverse procedure specified in Section 3.2 for screening differentially expressed genes. First, the FDR P values are computed for each gene at several MAD levels of interest. For each MAD level fcmin, we plot the ordered FDR P values. These can be plotted on the same gene index axis since the induced gene ordering is independent of MAD level. FDR P value curves for four different levels of fcmin are illustrated in Figure 4. The figure also illustrates how for FDR and MAD constraints $\alpha = 0.2$ and fcmin $= 0.32$, respectively, the G_2 positive responses $\mathcal{G}_2$ can be extracted from the FDR P value curve by thresholding. Notice that for fixed α, the size G_2 decreases rapidly as the MAD criterion becomes more stringent, that is, as fcmin increases.

Figure 5 shows nine of the top ranked (in FDR P value) differentially expressed gene profiles in (log base 2 scale) among the 59 genes selected by either the direct or inverse implementations of the FDR-CI screening procedure. In the figure, the level of significance constraint is FDR $\le \alpha = 0.2$ and the minimum foldchange constraint is MAD > fcmin $= 1.0$.

In Table 3, we compare the performance of the proposed screening algorithm, labeled "Two-stage FDR-CI," to two other algorithms, called "Thresholded FDR" and "Thresholded RMA." All three algorithms aim to control MAD at a level of fcmin $= 1.0$ (log base 2). The "Two-stage FDR-CI" and "Thresholded FDR" algorithms aim to control FDR at a level of $\alpha = 0.2$ in addition to MAD. Both of these latter algorithms were implemented as two-stage algorithms with common Stage 1, which is to select the gene responses $g \in \mathcal{G}_1$ that pass the paired-t test of hypotheses (1) with fcmin $= 0$ at a FDR level of 20%. The second stage of the "Two-stage FDR-CI" algorithm selects $\mathcal{G}_2$ as a subset of $\mathcal{G}_1$ at the prescribed FDR-CI level of 20%. Stage 2 of the "Thresholded FDR" algorithm simply selects the subset of genes $g \in \mathcal{G}_1$ having at least one sample mean foldchange exceeding fcmin $= 1.0$, that is, it implements the following filter:

$$\max_{t=1,2,3} |\overline{W_t}(g) - \overline{K_t}(g)| > 1.0 \tag{12}$$

on probes $g \in \mathcal{G}_1$. The single-stage "Thresholded RMA" algorithm, a nonstatistical method commonly used in many microarray studies, implements the filter (12) on the responses of each g in the original set of 12 421 genes as indicated in Figure 2.

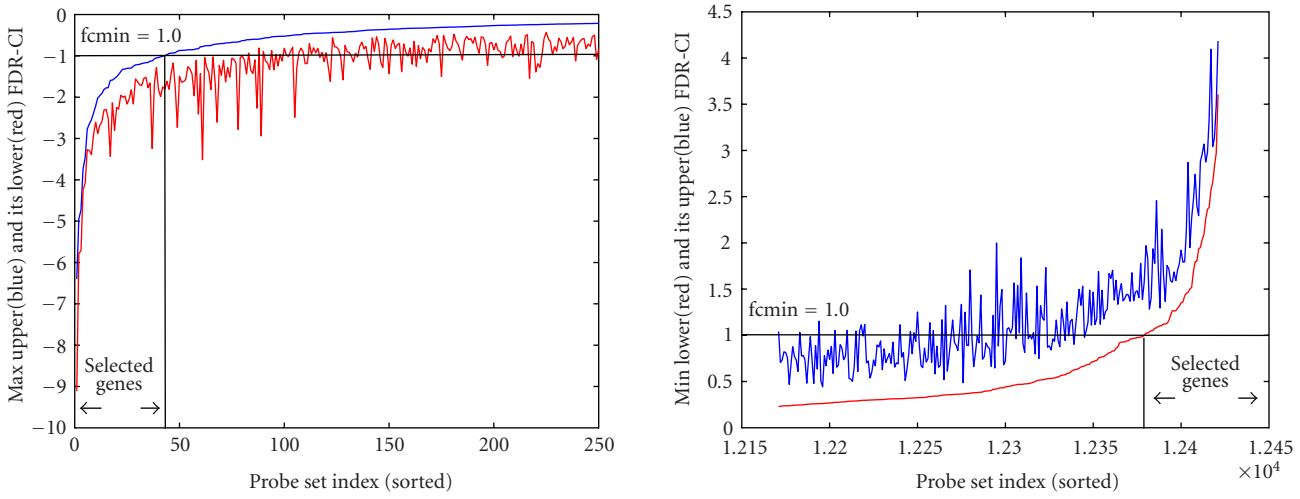


FIGURE 3: Segments of upper and lower curves specifying the 80% FDR-CI on the foldchanges $\{\mu_t(g) - \eta_t(g)\}_{t=1,2,3}$ for the knockout versus wild-type study. Upper and lower curves in each figure sweep out FDR-CI upper and lower boundaries on foldchange for all genes (indexed by probe set number). In (a) the curves sweep out the sequence of FDR-CIs indexed in an increasing order of the (maximum) lower CI boundary and in (b) the ordering is in an increasing order of the (minimum) upper CI boundary. Only those genes whose three FDR-CIs do not intersect $[-\text{fcmin}, \text{fcmin}]$ are selected by the second stage of screening. When the MAD foldchange criterion is fcmin = 2.0 (1.0 in log base 2), these genes are obtained by thresholding the curves as indicated.

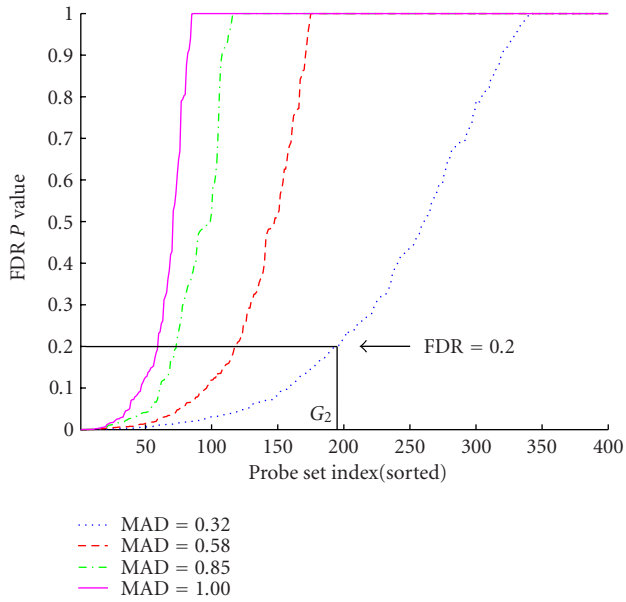


FIGURE 4: Plots of FDR P value curves over sorted list of gene indices for four values of the MAD criterion: fcmin = 0.32, 0.58, 0.85, 1.0 (log base 2) corresponding to wild-type/knockout MAD ratios of 1.25, 1.5, 1.8, and 2.0, respectively. Constraints FDR $\leq$ 0.2 and foldchange > 0.32 determine a set $\mathcal{G}_2$ of G_2 differentially expressed genes by thresholding the corresponding curve as indicated.

The number of screened and discovered genes for the three algorithms is indicated in the first two columns of Table 3. The maximum and median of the FDR P values of the discovered genes is indicated in the third and fourth columns for each algorithm. The last column indicates the maximum length of the FDR-CIs on foldchanges of the discovered genes. We conclude from Table 3 that the proposed "Two-stage FDR-CI" algorithm outperforms the other algorithms in terms of (1) maintaining the FDR requirement that false positives do not exceed 20% (column 4); (2) ensuring a substantially lower median FDR P value than the others (column 5); (3) discovering genes that have tighter (on the average) CIs on biologically significant (> 1.0) foldchange (column 6).

5. CONCLUSION

Signal processing for analysis of DNA microarrays for gene expression profiling is a rapidly growing area and there are enough challenges to keep the community busy for years. It is essential that signal processing methods be relevant and capture the biological aims of the experimenter. To this aim, in this paper, we developed a flexible multicriteria approach to gene selection and ranking for screening differentially expressed gene profiles. The proposed criteria capture the gene expression differences at multiple time points, account for minimum acceptable foldchange constraints, and control false discovery rate. In many cases, biological significance requires minimum hybridization levels, for example, as implemented by Affymetrix in their "absent calls" for weakly expressed genes. This can be easily captured by incorporating an addition criterion, the minimum acceptable mean expression level, into our multicriteria approach.

Table 3: Performance comparison of three algorithms for selecting genes with magnitude (log base 2) foldchange > 1.0. Thresholded RMA and Thresholded FDR are significantly worse in terms of statistical significance (P value) than the proposed Two-stage FDR-CI algorithm (columns 4 and 5). Furthermore, the average length of the CIs on foldchanges of the discovered genes are shorter for the Two-Stage FDR-CI algorithm than for the other algorithms (column 6).

	# Screened	# Discovered	Max(Pv)	Median(Pv)	Avg(FDR-CI length)
Thresholded RMA	12,421	159	1.0	0.80	1.52
Thresholded FDR	303	127	1.0	0.31	1.17
Two-stage FDR-CI	303	59	0.19	0.02	1.09

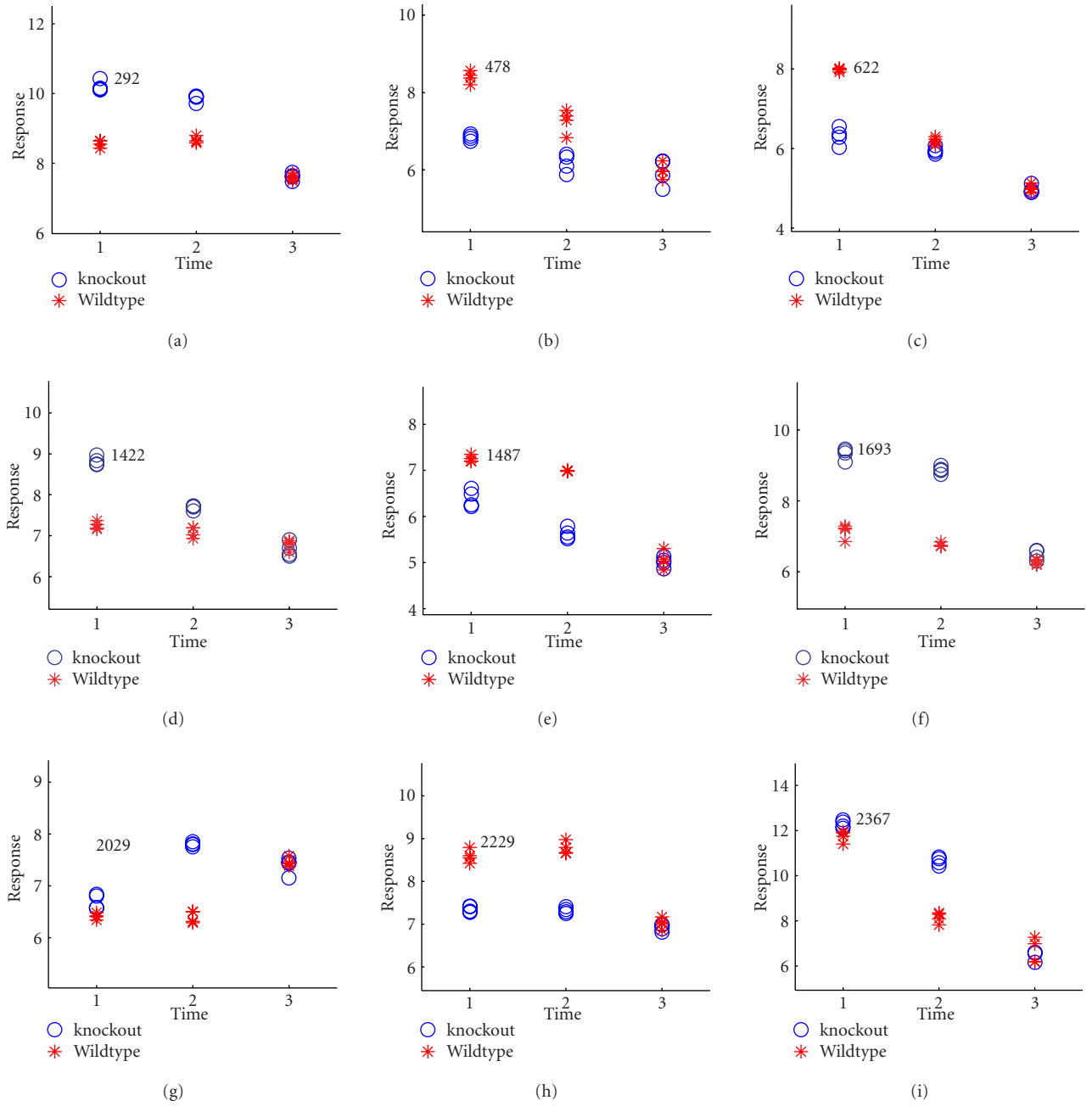


Figure 5: Gene profiles of nine of the differentially expressed genes discovered using the proposed two-stage FDR-CI procedure with constraints on level of significance $\alpha = 0.2$ and minimum foldchange fcmin = 1.0. Knockout "$\circ$" and Wildtype "$*$" are as indicated, and the numbers on each panel denote gene indices (related to the positions of the gene probes on the microarray).

ACKNOWLEDGMENT

The authors would like to thank R. Farjo for stimulating discussions and suggestions on the gene selection techniques presented in this paper. The research was supported by grants from the National Institutes of Health (EY11115 including administrative supplements), the Elmer and Sylvia Sramek Foundation, and The Foundation Fighting Blindness.

REFERENCES

[1] J. Watson and A. Berry, *DNA: The Secret of Life*, Alfred A. Knopf, NY, USA, 2003.

[2] F. C. Collins, M. Morgan, and A. Patrinos, "The Human Genome Project: lessons from large-scale biology," *Science*, vol. 300, no. 5617, pp. 286–290, 2003.

[3] P. O. Brown and D. Botstein, "Exploring the new world of the genome with DNA microarrays," *Nature Genetics*, vol. 21, suppl. 1, pp. 33–37, 1999.

[4] D. Bassett, M. B. Eisen, and M. Boguski, "Gene expression informatics—it's all in your mine," *Nature Genetics*, vol. 21, suppl. 1, pp. 51–55, 1999.

[5] Affymetrix, *NetAffx User's Guide*, 2000, http://www.netaffx.com/site/sitemap.jsp.

[6] National Human Genome Research Institute (NHGRI), *cDNA Microarrays*, 2001, http://www.nhgri.nih.gov/.

[7] T. Hastie, R. Tibshirani, M. Eisen, et al., "Gene shaving: a new class of clustering methods for expression arrays," Tech. Rep., Stanford University, Stanford, Callif, USA, 2000.

[8] A. A. Alizadeh, M. B. Eisen, R. E. Davis, et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[9] M. Brown, W. N. Grundy, D. Lin, et al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of National Academy of Sciences*, vol. 97, no. 1, pp. 262–267, 2000.

[10] Y. Benjamini and D. Yekutieli, "False discovery rate adjusted confidence intervals for selected parameters," submitted to Journal of the American Statistical Association.

[11] J. Liu and H. Iba, "Selecting informative genes using a multiobjective evolutionary algorithm," in *Proc. Congress on Evolutionary Computation*, pp. 297–302, Honolulu, Hawaii, USA, May 2002.

[12] G. Fleury, A. O. Hero, S. Yoshida, T. Carter, C. Barlow, and A. Swaroop, "Clustering gene expression signals from retinal microarray data," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 4, pp. 4024–4027, Orlando, Fla, USA, May 2002.

[13] A. Hero and G. Fleury, "Pareto-optimal methods for gene analysis," to appear in Journal of VLSI Signal Processing, Special Issue on Genomic Signal Processing.

[14] G. Fleury and A. O. Hero, "Gene discovery using Pareto depth sampling distributions," to appear in Journal of Franklin Institute.

[15] A. Reiner, D. Yekutieli, and Y. Benjamini, "Identifying differentially expressed genes using false discovery rate controlling procedures," *Bioinformatics*, vol. 19, no. 3, pp. 368–375, 2003.

[16] R. L. Miller, A. Galecki, and R. J. Shmookler-Reis, "Interpretation, design, and analysis of gene array expression experiments," *Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, vol. 56, no. 2, pp. B52–B57, 2001.

[17] D. B. Allison and C. S. Coffey, "Two stage testing in microarray analysis: what is gained?," *Journal of Gerontology: Biological Sciences*, vol. 57, no. 5, pp. B189–B192, 2002.

[18] Y. Benjamini, A. Krieger, and D. Yekutieli, "Adaptive linear step-up false discovery rate controlling procedures," Tech. Rep. Research Paper 01-03, Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv, Israel, 2001.

[19] T. P. Speed, *Statistical Analysis of Gene Expression Microarray Data*, Chapman & Hall/CRC Press, Boca Raton, Fla, USA, 2003.

[20] J. Watson, M. Gilman, J. Witkowski, and M. Zoller, *Recombinant DNA*, W. H. Freeman, NY, USA, 1992.

[21] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*, John Wiley & Sons, NY, USA, 2nd edition, 1999.

[22] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*, Holden-Day, San Francisco, Calif, USA, 1977.

[23] H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Part I*, John Wiley & Sons, NY, USA, 1968.

[24] S. Dudoit, J. P. Shaffer, and J. C. Boldrick, "Multiple hypothesis testing in microarray experiments," Tech. Rep. Working Paper 110, Berkeley Division of Biostatistics Working Paper Series, 2002, http://www.bepress.com/ucbbiostat/paper110.

[25] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society*, vol. 57, no. 1, pp. 289–300, 1995.

[26] R. G. Miller, *Simultaneous Statistical Inference*, Springer-Verlag, NY, USA, 1981.

[27] J. D. Storey and R. Tibshirani, "Estimating the positive false discovery rates under dependence, with applications to DNA microarrays," Tech. Rep. 2001-28, Department of Statistics, Stanford University, Stanford, Callif, USA, 2001.

[28] C. R. Genovese, N. A. Lazar, and T. E. Nichols, "Thresholding of statistical maps in functional neuroimaging using the false discovery rate," *NeuroImage*, vol. 15, no. 4, pp. 870–878, 2002.

[29] P. Westfall and S. Young, *Resampling-Based Multiple Testing*, John Wiley & Sons, NY, USA, 1993.

[30] V. S. Williams, L. V. Jones, and J. W. Tukey, "Controlling error in multiple comparisons, with examples from state-to-state differences in educational achievement," *Journal of Educational and Behavioral Statistics*, vol. 24, no. 1, pp. 42–69, 1999.

[31] A. Swaroop, J. Xu, H. Pawar, A. Jackson, C. Skolnick, and N. Agarwal, "A conserved retina-specific gene encodes a basic motif/leucine zipper domain," *Proceedings of National Academy of Sciences (USA)*, vol. 89, no. 1, pp. 266–270, 1992.

[32] A. Mears, M. Kondo, P. Swain, et al., "Nrl is required for rod photoreceptor development," *Nature Genetics*, vol. 29, no. 4, pp. 447–452, 2001.

[33] R. Irizarry, B. Hobbs, F. Collin, et al., "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, no. 2, pp. 249–264, 2003.

[34] D. F. Morrison, *Multivariate Statistical Methods*, McGraw-Hill, NY, USA, 1967.

Alfred O. Hero received his Ph.D. degree from Princeton University in 1984. Since then, he has been a Professor with the University of Michigan, Ann Arbor, where he has appointments in the Department of Electrical Engineering and Computer Science, the Department of Biomedical Engineering, and the Department of Statistics. Alfred Hero is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE). He has received the 1998 IEEE Signal Processing Society Meritorious Service Award, the 1998 IEEE Signal Processing Society Best Paper Award, and the IEEE Third Millenium Medal. His interests are in estimation and detection, statistical communications, bioinformatics, signal processing, and image processing.

Gilles Fleury was born in Bordeaux, France in 1968. He received the M.S. degree in electrical engineering from Ecole Supérieure d'Electricite (SUPELEC) in 1990, the Ph.D. degree in signal processing from the Université de Paris-Sud, Orsay, France, in 1994, and his Habilitation à diriger la Recherche (HDR) in 2003. He is presently a Professor within the Department of Measurement of SUPELEC. He has worked in the areas of inverse problems and optimal design. His current research interests include bioinformatics, optimal nonlinear modeling, and nonuniform sampling.

Alan J. Mears received his B.S. degree (Honors) from Leeds University, U.K. in 1989 and his Ph.D. degree from the University of Alberta, Canada in 1995, both in Genetics. He was a Research Investigator at the University of Michigan from 1999 to 2003 and is currently an Assistant Professor in Ophthalmology at the University of Ottawa in Canada. His research interests include the genetics of retinal disease and the transcriptional regulation of mammalian retinal development. Alan Mears has been a member of the American Association for the Advancement of Science from 1995 to 1997, American Society of Human Genetics from 1995 to 1998, and the Association for Research in Vision and Ophthalmology from 1996 till now.

Anand Swaroop received his Ph.D. degree in biochemistry from the Indian Institute of Science in 1982 and pursued his postdoctoral research in Genetics at Yale University, initially working on Drosophila and then Human Genetics. He joined the faculty of the Department of Ophthalmology and Visual Sciences and the Department Human Genetics at the University of Michigan Medical School in July 1990. He was promoted to a Full Professor in 2000 and currently holds the appointment as Harold F. Falls Collegiate Professor. He is Director/Coordinator of the Center for Retinal and Macular Degeneration and Director of the Sensory Gene Microarray Node. His research focuses on molecular genetics of retinal and macular diseases, retinal differentiation and aging, and expression profiling. He has published over 100 manuscript. His work is supported by grants from the National Institutes of Health, The Foundation Fighting Blindness, Macula Vision Research Foundation, and Elmer and Sylvia Sramek Charitable Foundation. In 1997, Anand Swaroop received the Lew R. Wasserman Merit Award from the Research to Prevent Blindness Foundation. He is currently a member on the editorial boards of Investigative Ophthalmology and Visual Science and Molecular Vision. He reviews manuscripts and grants for several journals, international foundations, and agencies. He is also a regular member of the BDPE study section of NIH.

EURASIP Journal on Applied Signal Processing 2004:1, 53–63

The Local Maximum Clustering Method and Its Application in Microarray Gene Expression Data Analysis

Xiongwu Wu
Laboratory of Biophysical Chemistry, National Heart, Lung, and Blood Institute, National Institutes of Health, Bethesda, MD 20892, USA
Email: wuxw@nhlbi.nih.gov

Yidong Chen
National Human Genome Research Institute, National Institutes of Health, Bethesda, MD 20892, USA
Email: yidong@nhgri.nih.gov

Bernard R. Brooks
Laboratory of Biophysical Chemistry, National Heart, Lung, and Blood Institute, National Institutes of Health, Bethesda, MD 20892, USA
Email: brb@nih.gov

Yan A. Su
Department of Pathology, Loyola University Medical Center, Maywood, IL 60153, USA
Email: ysu2@lumc.edu

Received 28 February 2003; Revised 25 July 2003

An unsupervised data clustering method, called the local maximum clustering (LMC) method, is proposed for identifying clusters in experiment data sets based on research interest. A magnitude property is defined according to research purposes, and data sets are clustered around each local maximum of the magnitude property. By properly defining a magnitude property, this method can overcome many difficulties in microarray data clustering such as reduced projection in similarities, noises, and arbitrary gene distribution. To critically evaluate the performance of this clustering method in comparison with other methods, we designed three model data sets with known cluster distributions and applied the LMC method as well as the hierarchic clustering method, the K-mean clustering method, and the self-organized map method to these model data sets. The results show that the LMC method produces the most accurate clustering results. As an example of application, we applied the method to cluster the leukemia samples reported in the microarray study of Golub et al. (1999).

Keywords and phrases: data cluster, clustering method, microarray, gene expression, classification, model data sets.

1. INTRODUCTION

Data analysis is a key step in obtaining information from large-scale gene expression data. Many analysis methods and algorithms have been developed for the analysis of the gene expression matrix [1, 2, 3, 4, 5, 6, 7, 8, 9]. The clustering of genes for finding coregulated and functionally related groups is particularly interesting in cases where there is a complete set of organism's genes. A reasonable hypothesis is that genes with similar expression profiles, that is, genes that are co-expressed, may have something in common in their regulatory mechanisms, that is, they may be coregulated. Therefore, by clustering together genes with similar expression profiles, one can find groups of potentially coregulated genes and search for putative regulatory signals. So far, many clustering methods have been developed. They can be divided into two categories: supervised and unsupervised methods. This work focuses on unsupervised data clustering. Some widely used methods in this category are the hierarchic clustering method [6], the K-mean clustering method [10], and the self-organized map clustering method [9, 11].

The clustering of microarray gene expression data typically aims to group genes with similar biological functions or to classify samples with similar gene expression profiles. There are several factors that make the clustering of gene expression data different from data clustering in a general

sense. First, the "positions" of genes or samples are unknown. That is, where the data points to be clustered locate is unknown. Instead, the relations between data points (genes or samples) are probed by a series of responses (gene expressions). Generally, the correlation of the response series between data points is used as a measure of their similarity. However, because the number of responses is limited and the responses are not independent from each other, the correlation can only provide a reduced description of the similarities between data points. Just like a projection of data points in a high-dimensional space to a low-dimensional space, many data points far apart may be projected together. It often happens that genes that belong to very different categories are clustered together according to gene expression data. Second, there is only a small number of genes presented in a microarray that are relevant to the biological processes under study. All the rest become noises to the analysis, which need to be filtered out based on some criteria before clustering analysis. Third, the genes chosen to array do not necessarily represent the functional distribution. That is, there exist redundant genes of some functions while very few genes exist of some other functions. This may result in the neglect of those less-redundant gene clusters in a clustering analysis. These facts rise difficulties and uncertainties for cluster analysis. Fortunately, a microarray experiment does not attempt to provide accurate cluster information of all genes being arrayed. Instead, besides many other purposes, a microarray experiment is designed to identify and study those groups, which seem to participate in the studied biological process. The complete gene cluster will be the job of many molecular biology experiments as well as other technologies.

With our interest focused on those functional related genes, we need to identify clusters functionally relevant to the biological process of interest. As stated above, clustering methods solely dependent on similarities may suffer from the difficulties of reduced projection, noises, and arbitrary gene distribution and may not be suitable for microarray research purposes. In this work, we present a general approach to clustering a data set based on research interest. A quantity, which is generally called magnitude, is introduced to represent a property of our interest for clustering. The following sections explain in detail the concept and the clustering method, which we call the local maximum clustering (LMC) method. Additionally, for the purpose of comparison, we worked out an approach to quantitatively calculate the agreement between two hierarchic clustering results for the same data set. Using three model systems, we compared this clustering method with several well-known clustering methods. Finally, as an example of application, we applied the method to cluster the leukemia samples reported in the microarray study of Golub et al. [12].

2. METHODS AND ALGORITHMS

2.1. Distances, magnitudes, and clusters

For a data set with unknown absolute positions, the distance matrix between data points is used to infer their relative positions. For a biologically interesting data set like genes or tissue samples, the distances are not directly measurable. Instead, the responses to a series event are used to estimate the distances or similarity. It is assumed that data points close to each other have similar responses.

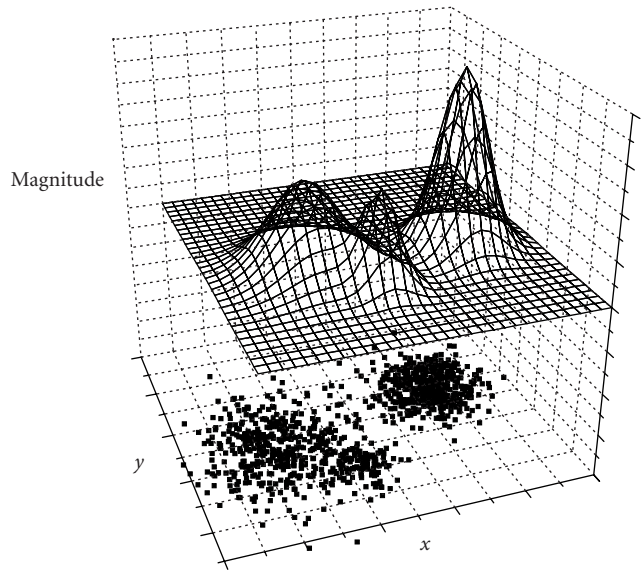


Figure 1: A two-dimensional (x-y) distribution data set with the "magnitude" as the additional dimension.

For microarray gene expression data, people often use Pearson correlation function to describe the similarity between genes i and j:

$$C_{ij} = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{X_{ik}-\overline{X}_i}{\sigma_i}\right)\left(\frac{X_{jk}-\overline{X}_j}{\sigma_j}\right), \tag{1}$$

where $X_i = (X_{ik})_n$, $k = 1,\ldots,n$, represents the data point of gene i, which consists of n responses, X_{ik} is the kth response of gene i, $\overline{X}_i$ is the average value of X_i, $\overline{X}_i = (1/n)\sum_{k=1}^{n} X_{ik}$, and σ_i is the standard deviation of X_i, $\sigma_i = \sqrt{\overline{X_i^2} - \overline{X}_i^2}$.

From (1), we can see that C_{ij} ranges from -1 to 1, with 1 representing identical responses between genes i and j and -1 the opposite responses. The distance between a pair of genes is often expressed as the following function:

$$r_{ij} = 1 - C_{ij}. \tag{2}$$

We introduce a quantity called magnitude to represent our research interest. This magnitude is introduced as an additional dimension to the distribution space. We image a set of data points distributed on x-y plan, a two-dimensional space, the magnitude will be an additional dimension, z-dimension (Figure 1). Usually, a cluster is a collection of data points that are more similar to each other than to data points in different clusters. Clusters of this type are characterized by a magnitude of the local densities with each cluster representing a high-density region. Here, the local density is the

magnitude used to define clusters. We should keep in mind that the magnitude property can be properties other than density; it can be gene expression levels or gene differential expressions as described later. As can be seen from Figure 1, each cluster is represented by a peak on the magnitude surface. Obviously, clusters in a data set can be found out by identifying peaks on the magnitude surface. Because clusters are peaks on the magnitude surface, the number and size of clusters depend only on the surface shape.

Current existing clustering methods like the hierarchic clustering method do not explicitly use the magnitude property. These clustering methods assume clusters locate at high-density areas of a distribution. In other words, these clustering methods implicitly use distribution density as the magnitude of clustering.

The choosing of the magnitude property determines what we want to be the cluster centers. If we want clusters to center at high-density areas, using distribution density would be a natural choice for the magnitude. A simple distribution density can be calculated as

$$M_i = \sum_{j=1}^{n} \delta(r_{ij}), \tag{3}$$

where $\delta(r_{ij})$ is a step function:

$$\delta(r_{ij}) = \begin{cases} 1 & r_{ij} \le d \\ 0 & r_{ij} > d. \end{cases} \tag{4}$$

Equation (3) indicates the magnitude of data point i and M_i is equal to the number of data points within distance d from data point i. A smaller d will result in a more accurate local density but a larger statistic error. To make the magnitude smooth, an alternative function can be used for $\delta(r_{ij})$:

$$\delta(r_{ij}) = \exp\left(-\frac{r_{ij}^2}{2d^2}\right). \tag{5}$$

For microarray studies, directly clustering genes based on density may result in misleading results. The main reason is that we do not know the real "positions" of the genes. The relative similarities between genes are probed by their responses to an often very limited number of samples. The similarity obtained this way is a reduced projection of "real" similarities, and many very different functional genes may respond similarly in the limited sample set. Therefore, the densities estimated from the response data are not reliable and change from experiment to experiment. Further, the correlation function captures similarity of the shapes of two expression profiles, but it ignores the strength of their responses. Some noises in response measurement may cause a nonresponsive gene to be of high correlation with a high-response gene. Another reason is that the genes arrayed in a chip may vary in redundancy, resulting in different density distributions. An extreme case is when a single gene is redundant so many times that they occupy a large portion of an array—a cluster centering at this gene would be created. Additionally, for the thousands of genes arrayed on a gene chip, generally, only a handful of genes show varying expression levels, which we used to probe gene functions. All the rest only show undetectable expressions or simply noises which may result in very high correlation to some genes. Normally, only those genes with significantly varying expression levels can be of meaningfully functional relation, while for the rest we can draw little information from a microarray experiment. Therefore, for a microarray study, a good choice of magnitude would be a quantity measuring the variation of expression levels as in

$$M_i = \delta^2(\ln R_i) = \frac{1}{n}\sum_{j=1}^{n}(\ln R_{ij})^2 - \left(\frac{1}{n}\sum_{j=1}^{n}\ln R_{ij}\right)^2, \tag{6}$$

where R_i is the expression ratio between sample and control and n is the number of samples for each gene. Equation (6) is a magnitude defined as the differential expression of genes. By this definition, the clusters are always centered at high-differential expression genes. Because this paper focuses on the presentation and evaluation of the local maximum clustering method, we will not discuss the application of (6) in identifying high-response gene clusters. This equation is presented here only to illustrate the idea of the magnitude properties.

2.2. The local maximum clustering method

Two types of properties characterize the data points: magnitude of each data point and distance (or similarity) between a pair of data points. We define a cluster as a peak on the magnitude surface. Therefore, we can cluster a data set by identifying peaks on the magnitude surface.

There are many approaches to identifying peaks on a surface. Here, in this work, we use a method called the local maximum method to identify peaks. Identification of peaks on a surface can be done by searching for the local maximum point around each data point. Assume there is a data set of N data points to be clustered. The local maximum of a data point i is the data point whose magnitude is the maximum among all the data points within a certain distance from the data point i. A peak has the maximum magnitude in its local area, therefore, its local maximum is itself. By identifying all data points whose local maximum points are themselves, we can locate all the peaks on the magnitude surface. The distance used to define the local area is called resolution. The number of peaks on a magnitude surface depends on the shape of the surface and the size of resolution. After the peaks are identified, all data points can be assigned into these peaks according to their local maximum points in the way that a data point belongs to the same peak as its local maximum point.

Figure 2 shows a one-dimensional distribution of a data set along the x-axis. The y-axis is the magnitude of the data set. The peaks represent cluster centers depending on the resolution r_0. Clusters can be identified by searching for the peaks in the distribution, and all data points can be clustered into these peaks according to the local maximums of each data point. Assume that r_1, r_3, and r_4 are the distances from peaks 1, 3, and 4 to their nearest equal-magnitude neighbor points. With a resolution $r_0 < r_3$, four peaks, 1, 2, 3, and 4 can be identified as the local maximum points of themselves. All

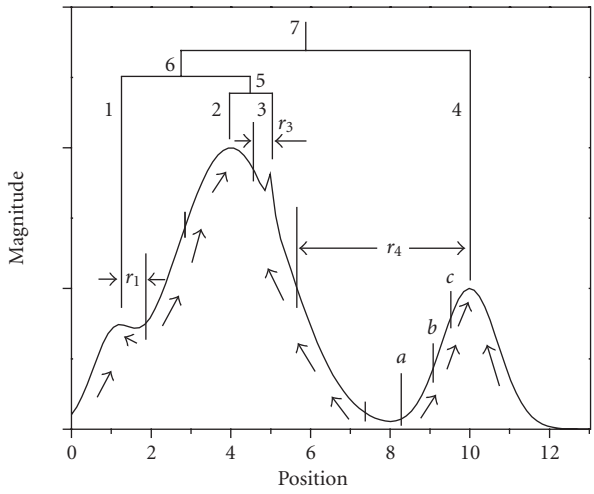


FIGURE 2: Clustering a data set based on the local maximum of its magnitude. There are 4 peaks, 1, 2, 3, and 4; and r_1, r_3, and r_4 are the distances from peaks 1, 3, and 4 to their nearest equal magnitude neighbor points. Assume $r_3 < r_1 < r_4$.

data points can be clustered into these four peaks according to their local maximum points. For example, for data point a, if data point b is the one that has the maximum magnitude in all data points within r_0 from a, we say b is the local maximum point of a. Point a will belong to the same peak as point b. Similarly, point b belongs to the same peak as its local maximum point c and point c belongs to peak 4. Therefore, points a, b, and c all belong to peak 4.

Obviously, resolution r_0 plays a crucial role in identifying peaks. For each peak p, we define its resolution limit r_p as the longest distance within which peak p has the maximum magnitude. For a given resolution r_0, a peak p will be identified as a cluster center if $r_p > r_0$. As shown in Figure 2, there are four peaks, 1, 2, 3, and 4. If $r_0 > r_1$, peak 1 will not be identified and, together with all its neighbors, will be assigned to cluster 2. Similarly, cluster 3 or 4 can only be identified when $r_0 < r_3$ or $r_0 < r_4$, respectively.

The peaks identified can be further clustered to produce a hierarchic cluster structure. For the example shown in Figure 2, if we assume that $r_4 > r_1 > r_3$, by using $r_0 < r_3$, we can get four clusters, while, using $r_1 > r_0 > r_3$, clusters 2 and 3 merge to cluster 5 at peak 2, with $r_4 > r_0 > r_1$, clusters 1 and 5 merge into cluster 6 at peak 2, and with $r_0 > r_4$, all clusters merge into a single cluster at peak 2.

The algorithm of the LMC method is described by the following steps.

(i) For a data set $\{i\}$, $i = 1, 2, \ldots, N$, calculate the distances between data points $\{r_{ij}\}$ using (1) and (2). From the distance matrix, calculate the magnitude of each data point $\{M(i)\}$ using (5).

(ii) Set resolution $r_0 = \min\{r_{ij}\} + \delta r$, $i \neq j$. Here, δr is the resolution increment. Typically, set $\delta r = 0.01$.

(iii) Search for the local maximum point $L(i)$ for each data point i. For all j, with $r_{ij} < r_0$, there is $M(L(i)) \geq M(j)$.

(iv) Identify peak centers $\{p\}$, where $L(p) = p$. Each peak represents the center of a cluster.

(v) Assign each data point i to the same cluster as its local maximum point $L(i)$.

(vi) If there is more than one cluster, generate higher-level clusters from the peak point data set $\{p\}$, $p = 1, 2, \ldots, n_p$, following steps (ii), (iii), (iv), and (v).

2.3. Comparison of hierarchic clusters

For the same data set, different clustering methods may produce different clusters. It is, in general, a nontrivial task to compare different clustering results of the same data set and many efforts have been made for such clustering comparison (e.g., [13]). For hierarchic clustering, comparison is more challenging because a hierarchic cluster is a cluster of clusters. To quantitatively compare hierarchic clusters from different methods, we define the following agreement function to describe the agreement between hierarchic clustering results.

We use $\{H_1\}$ and $\{H_2\}$ to represent two hierarchic clustering results for the same data set. In the following discussions, N_1 and N_2 are the numbers of clusters in $\{H_1\}$ and $\{H_2\}$, respectively, n_{1i} and n_{2j} represent the data point numbers in cluster i of $\{H_1\}$ and cluster j of $\{H_2\}$, respectively, and m_{ij} is the number of data points existing both in cluster i of $\{H_1\}$ and in cluster j of $\{H_2\}$. Therefore, $2m_{ij}/(n_{1i} + n_{2j})$ represents how well the two clusters, cluster i of $\{H_1\}$ and cluster j of $\{H_2\}$, are similar to each other. A value of 1 indicates they are identical and a value of 0 indicates they are completely different. We use $M_{1i}(\{H_2\})$ to describe how well cluster i of $\{H_1\}$ is clustered in $\{H_2\}$. We call $M_{1i}(\{H_2\})$ the match of $\{H_1\}$ to $\{H_2\}$ in cluster i. Similarly, the match of $\{H_2\}$ to $\{H_1\}$ in cluster j is denoted as $M_{2j}(\{H_1\})$, which describes how well cluster j of $\{H_2\}$ is clustered in $\{H_1\}$. They are calculated using the following equations:

$$M_{1i}(\{H_2\}) = \max_{j \in N_2} \left\{ \frac{2m_{ij}}{n_{1i} + n_{2j}} \right\},$$
$$M_{2j}(\{H_1\}) = \max_{i \in N_1} \left\{ \frac{2m_{ij}}{n_{1i} + n_{2j}} \right\}. \tag{7}$$

Equations (7) mean that the match of $\{H_1\}$ to $\{H_2\}$ in a cluster is the highest similarity between this cluster and any cluster of $\{H_2\}$.

We use the agreement $A(\{H_1\}, \{H_2\})$ to describe the overall similarity between two clustering results, which is a weighted average of all cluster matches, as

$$A(\{H_1\}, \{H_2\}) = \frac{1}{2\sum_{i=1}^{N_1} n_{1i}} \sum_{i=1}^{N_1} n_{1i} M_{1i}(\{H_2\}) + \frac{1}{2\sum_{j=1}^{N_2} n_{2j}} \sum_{j=1}^{N_2} n_{2j} M_{2j}(\{H_1\}). \tag{8}$$

To further illustrate the definition of the agreement and matches, we show an example of two hierarchic clustering results in Figures 3a and 3b. These two hierarchic clustering results, $\{H_A\}$ and $\{H_B\}$, are for the same data set of 1000

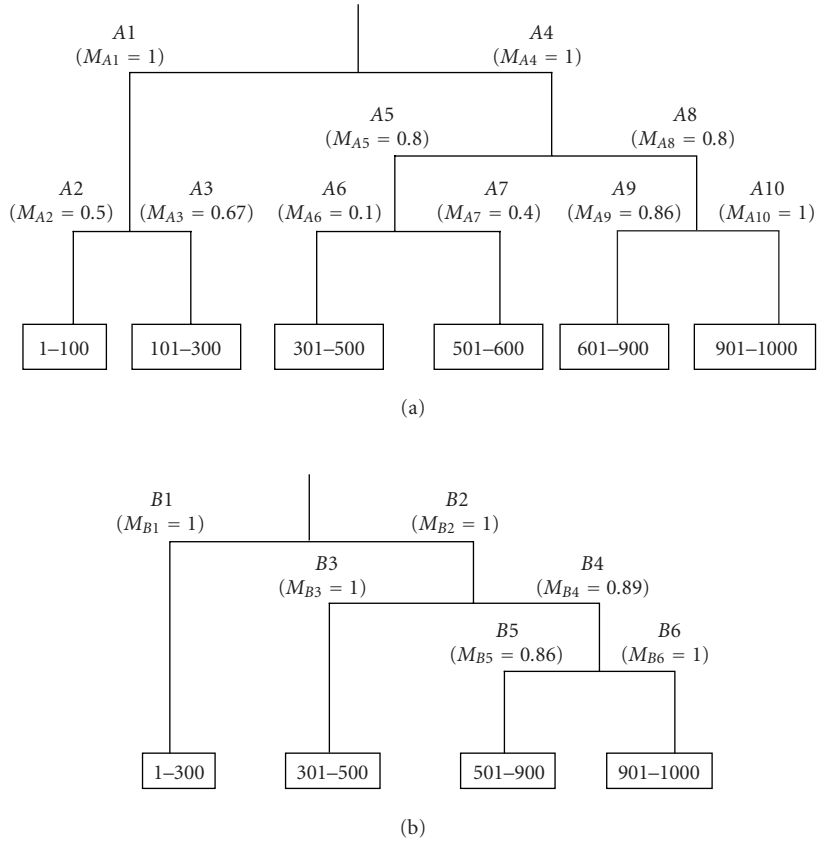


FIGURE 3: (a) The hierarchic clustering structure $\{H_A\}$ with 10 clusters; the match of each cluster to the cluster structure $\{H_B\}$ are labeled in parentheses; (b) the hierarchic cluster structure $\{H_B\}$ with 6 clusters; the match of each cluster to the cluster structure $\{H_A\}$ are labeled in parentheses.

data points. The hierarchic clustering structure $\{H_A\}$ has 10 clusters and $\{H_B\}$ has 6 clusters. Clusters $A1$, $A4$, and $A10$ of $\{H_A\}$ have the same data points as clusters $B1$, $B2$, and $B6$ of $\{H_B\}$, respectively. Therefore, their matches are 1 no matter how different their subclusters are. The matches of clusters are calculated according to (7) and are labeled in the figures. The agreement between $\{H_A\}$ and $\{H_B\}$ can be calculated using (8) as follows:

$$
\begin{aligned}
&A(\{H_A\},\{H_B\})\\
&\quad=\frac{\sum_{i=1}^{10} n_{Ai}M_{Ai}}{2\sum_{i=1}^{10} n_{Ai}}+\frac{\sum_{j=1}^{6} n_{Bj}M_{Bj}}{2\sum_{j=1}^{6} n_{Bj}}\\
&\quad=\frac{300\times 1+100\times 0.5+200\times 0.67+700\times 1+300\times 0.8+200\times 0.1+100\times 0.4+400\times 0.8+300\times 0.86+100\times 1}{2(300+100+200+700+300+200+100+400+300+100)}\\
&\qquad+\frac{300\times 1+700\times 1+200\times 1+500\times 0.89+400\times 0.86+100\times 1}{2(300+700+200+500+400+100)}\\
&\quad=0.400+0.475\\
&\quad=0.875.
\end{aligned}
\tag{9}
$$

Table 1: The possibility parameters used to generate the three model systems. Each model has 6 clusters. The parameters (h_i, w_i) represent the height and width of cluster i in the possibility distribution in (10).

Model	(h_1, w_1)	(h_2, w_2)	(h_3, w_3)	(h_4, w_4)	(h_5, w_5)	(h_6, w_6)
1	(1, 0.05)	(1, 0.02)	(1, 0.02)	(1, 0.05)	(1, 0.02)	(1, 0.02)
2	(1, 0.10)	(1, 0.005)	(1, 0.05)	(1, 0.10)	(1, 0.005)	(1, 0.10)
3	(1, 0.10)	(2, 0.005)	(3, 0.05)	(4, 0.10)	(5, 0.005)	(6, 0.10)

3. RESULTS AND DISCUSSIONS

The LMC method has several features. First, it is an unsupervised clustering method. The clustering result depends on the data set itself. Second, it allows magnitude properties to be used to identify clusters of interest. Third, it automatically produces a hierarchic cluster structure with a minimum amount of input. In this work, we designed three model systems with known cluster distributions to evaluate the performance of the LMC method and compare it with other methods. Finally, as an example of application, we use this method to cluster the leukemia samples reported by Golub et al. [12] and compare the result with experimental classification.

3.1. The model systems

Model systems with known cluster distributions have often been used in method development. The model systems used here are designed to mimic microarray gene expression data in the way that each data point is a response series of expression values, and the distance or similarity between data points is measured by their correlation function. It is the correlation function that determines the distance between data points and the actual number of expression values in a response series, which does not affect the clustering results; for simplicity and convenience of data generation and analysis, we use only three expression values for each response series, namely, x, y, and z. The response series of gene i is represented by (x_i, y_i, z_i). The correlation function and distance between gene i and gene j is calculated according to (1) and (2) with $n = 3$.

The model systems are designed to have 6 clusters with cluster centers at (X_j, Y_j, Z_j), $j = 1, 2, 3, 4, 5$, and 6. We use the following possibility distribution to generate the expression data of 1000 genes (x_i, y_i, z_i), $i = 1, 2, \ldots, 1000$:

$$\rho(x_i, y_i, z_i) = \sum_{j=1}^{6} h_j \exp\left(- \frac{(1 - C_{ij})^2}{2w_j^2} \right), \qquad (10)$$

where $\rho(x_i, y_i, z_i)$ represents the possibility function to have a gene with a response series of $\rho(x_i, y_i, z_i)$, and h_j and w_j are the height and width of cluster j. The six cluster centers are genes with the following response series:

(i) $(-\sqrt{2}/2, 0, \sqrt{2}/2)$;
(ii) $(-\sqrt{2}/2, \sqrt{2}/2, 0)$;
(iii) $(-1/\sqrt{6}, 2/\sqrt{6}, -1/\sqrt{6})$;
(iv) $(0, -\sqrt{2}/2, \sqrt{2}/2)$;

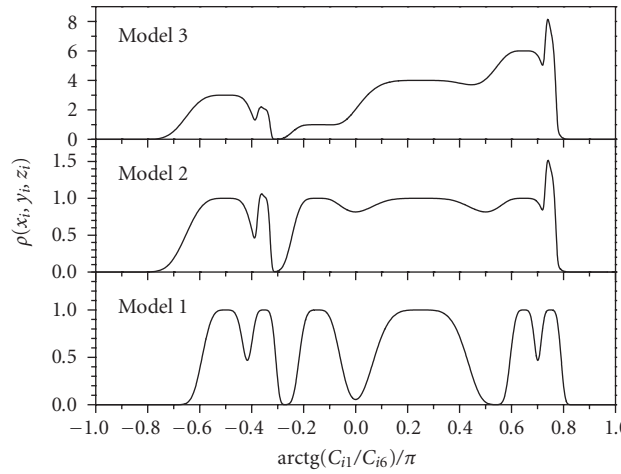


Figure 4: Data distribution in the three model data sets. The function arctg$(C_{i1}/C_{i6})/\pi$ is used for the x-axis to show all six clusters without overlapping. Here, C_{i1} and C_{i6} are the correlations of data point i with the centers of clusters 1 and 6, respectively. For each model, 1000 data points are generated.

(v) $(2/\sqrt{6}, -1/\sqrt{6}, -1/\sqrt{6})$;
(vi) $(\sqrt{2}/2, -\sqrt{2}/2, 0)$.

The correlation matrix between these centering genes is

$$[C_{ij}]_{6\times 6} = \begin{pmatrix} 1 & \frac{1}{2} & 0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & 1 & \frac{\sqrt{3}}{2} & -\frac{1}{2} & -\frac{\sqrt{3}}{2} & -1 \\ 0 & \frac{\sqrt{3}}{2} & 1 & -\frac{\sqrt{3}}{2} & -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 & 0 & \frac{1}{2} \\ -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 & 1 & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -1 & -\frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \end{pmatrix}. \qquad (11)$$

Three model data sets, each has 1000 data points, are generated using the parameters listed in Table 1. Their distributions are shown in Figure 4. The clusters are separated

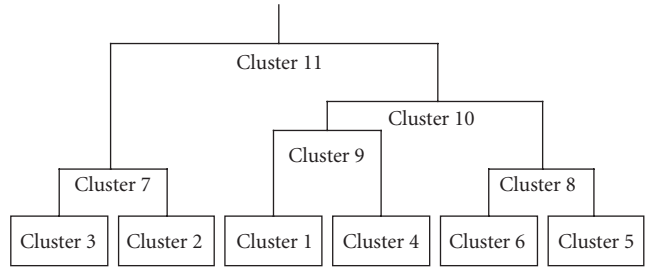


Figure 5: The hierarchic cluster structure of the model data sets.

Table 2: Comparison of the clustering results of different methods. The letters L, H, K, and S stand for the LMC method, the hierarchic clustering method, the K-mean clustering method, and the self-organization map clustering method, respectively.

		Model 1				Model 2				Model 3			
	Clusters	L	H	K	S	L	H	K	S	L	H	K	S
Matches to the models (%)	1	99.7	97.2	68.0	68.0	87.8	87.8	87.8	87.2	89.8	85.2	85.0	85.2
	2	99.2	96.8	65.2	65.2	98.0	94.6	35.6	36.0	78.2	85.8	41.0	40.8
	3	99.6	99.6	69.7	88.3	94.4	80.8	71.1	67.4	91.8	43.8	95.8	70.7
	4	99.8	99.8	68.8	77.4	69.5	67.5	77.5	72.3	89.0	78.8	71.8	70.8
	5	98.1	99.2	62.3	63.1	80.1	76.9	76.2	80.4	88.4	76.6	78.8	65.4
	6	98.4	98.4	70.6	70.2	92.5	96.9	70.0	45.2	91.1	96.2	75.8	55.8
	7	99.8	99.8	—	—	99.8	99.7	—	—	99.8	99.8	—	—
	8	100	100	—	—	97.2	95.0	—	—	95.1	94.4	—	—
	9	99.8	99.8	—	—	98.4	82.8	—	—	95.0	94.4	—	—
	10	100	76.8	—	—	100	100	—	—	100	100	—	—
Overall agreement (%)		96.9	69.4	76.2	81.0	88.5	65.1	75.3	76.0	89.5	67.2	79.5	72.9

by minimums between peaks, and the data points can be accurately assigned to their clusters. As can be seen (Figure 4) in model 1, the six clusters have equal heights and are clearly separated from each other, while in model 2, clusters 1, 3, 4, and 5 are much broader, and in model 3, their heights are different. These three model data sets present some typical cases that a clustering method would deal with.

Based on the correlations between the clusters, (11), these model data sets have a hierarchic cluster structure as shown in Figure 5. The whole data set belongs to a single cluster 11, which is split into two clusters, 7 and 10. Cluster 7 is divided into clusters 2 and 3. Cluster 10 is further divided into cluster 9, which consists of clusters 1 and 4, and cluster 8, which consists of clusters 5 and 6.

We applied the LMC method (L), the hierarchic clustering method [6] (H), the K-mean clustering method [10] (K), and the self-organized map clustering method [11] (S) to these three model data sets. The LMC method, as well as the hierarchic clustering method, produces a hierarchic cluster structure. The K-mean and the self-organized map methods require a predefined cluster number prior to clustering. For comparison purpose, we set the cluster number to 6 when performing clustering using the K-mean and the self-organized map method, and only compare the agreement between the clustering results with the bottom 6 clusters of the model data sets. Table 2 listed the matches and agreements between the results from the four clustering methods and the known clusters of the model data sets.

Comparing the matches and agreements between the clustering results and the known clusters of the model data sets, we can see clearly that the LMC method produces the most accurate result. The hierarchic clustering method produces many tree structures, within which there exist good matches to the clusters in the models. Because it produces too many trees, the agreement between the model and result from the hierarchic method is low. The K-mean and the self-organized map methods produce worse matches to the clusters in the models than the LMC and the hierarchic clustering methods.

3.2. An application to microarray gene expression data

Application of the LMC method to gene expression data is straightforward. As an example of the application, we applied this method to cluster the 72 samples collected by Golub et

Table 3: Classification of the acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) samples [12].

Cluster levels				Samples	Type	Source	Lineage	FAB	Sex
1	2	3	4						
A	*A*1	*A*11	*A*111	4	ALL	BM	B-cell	—	—
				20	ALL	BM	B-cell	—	—
				5	ALL	BM	B-cell	—	—
				19	ALL	BM	B-cell	—	—
			*A*112	46	ALL	BM	B-cell	—	F
				12	ALL	BM	B-cell	—	F
				42	ALL	BM	B-cell	—	F
				48	ALL	BM	B-cell	—	F
				7	ALL	BM	B-cell	—	F
				59	ALL	BM	B-cell	—	F
				8	ALL	BM	B-cell	—	F
				15	ALL	BM	B-cell	—	F
				18	ALL	BM	B-cell	—	F
				43	ALL	BM	B-cell	—	F
				56	ALL	BM	B-cell	—	F
				40	ALL	BM	B-cell	—	F
				44	ALL	BM	B-cell	—	F
				27	ALL	BM	B-cell	—	F
				26	ALL	BM	B-cell	—	F
				55	ALL	BM	B-cell	—	F
				39	ALL	BM	B-cell	—	F
				41	ALL	BM	B-cell	—	F
				13	ALL	BM	B-cell	—	F
			*A*113	17	ALL	BM	B-cell	—	M
				16	ALL	BM	B-cell	—	M
				21	ALL	BM	B-cell	—	M
				45	ALL	BM	B-cell	—	M
				22	ALL	BM	B-cell	—	M
				25	ALL	BM	B-cell	—	M
				24	ALL	BM	B-cell	—	M
				47	ALL	BM	B-cell	—	M
				1	ALL	BM	B-cell	—	M
				49	ALL	BM	B-cell	—	M
		*A*12		23	ALL	BM	T-cell	—	M
				10	ALL	BM	T-cell	—	M
				3	ALL	BM	T-cell	—	M
				11	ALL	BM	T-cell	—	M
				2	ALL	BM	T-cell	—	M
				6	ALL	BM	T-cell	—	M
				14	ALL	BM	T-cell	—	M
				9	ALL	BM	T-cell	—	M
	*A*2	*A*21	*A*211	72	ALL	PB	B-cell	—	—
				71	ALL	PB	B-cell	—	—
			*A*212	70	ALL	PB	B-cell	—	F
			*A*213	68	ALL	PB	B-cell	—	M
				69	ALL	PB	B-cell	—	M
		*A*22		67	ALL	PB	T-cell	—	M

Table 3: Continued.

Cluster levels				Samples	Type	Source	Lineage	FAB	Sex
1	2	3	4						
B	*B*1	*B*11		66	AML	BM	—	—	M
				65	AML	BM	—	—	M
		*B*12		35	AML	BM	—	*M*1	—
				38	AML	BM	—	*M*1	—
				61	AML	BM	—	*M*1	—
				32	AML	BM	—	*M*1	—
		*B*13	*B*131	58	AML	BM	—	*M*2	—
				34	AML	BM	—	*M*2	—
				28	AML	BM	—	*M*2	—
				37	AML	BM	—	*M*2	—
				51	AML	BM	—	*M*2	—
				29	AML	BM	—	*M*2	—
				33	AML	BM	—	*M*2	—
				53	AML	BM	—	*M*2	—
			*B*132	57	AML	BM	—	*M*2	F
			*B*133	60	AML	BM	—	*M*2	M
		*B*14	*B*141	31	AML	BM	—	*M*4	—
				50	AML	BM	—	*M*4	—
			*B*142	54	AML	BM	—	*M*4	F
		*B*15		36	AML	BM	—	*M*5	—
				30	AML	BM	—	*M*5	—
	*B*2	*B*21	*B*211	63	AML	PB	—	—	F
			*B*212	64	AML	PB	—	—	M
				62	AML	PB	—	—	M
		*B*22		52	AML	PB	—	*M*4	—

al. [12] from acute leukemia patients at the time of diagnosis. We choose this data because experimental classification is available for comparison. Table 3 lists the clusters based on experiment classification [12]. The 72 samples contain 47 acute lymphoblastic leukemia (ALL) samples (cluster A) and 25 acute myeloid leukemia (AML) samples (cluster B). These samples are from either bone marrow (BM) (clusters A1 and B1) or peripheral blood (PB) (clusters A2 and B2). The ALL samples fall into two classes: B-lineage ALL (clusters A11 and A21) and T-lineage ALL (clusters A12 and A22), some of which are taken from known sex patients (F for female and M for male). Some of the AML samples have known FAB types, M1–M5.

The whole set of genes are filtered based on expression levels, and 1769 genes with expression levels higher than 20 in all the 72 samples are used for our clustering. That is, for each sample, its response series contains 1769 gene expression values. The logarithms of the gene expression levels are used in correlation function calculation to reduce the noise effect at high expression levels.

We applied the LMC method and the hierarchic clustering method [6] to the 72 samples and compared the results with the experiment clusters listed in Table 3. The magnitude is calculated using (5) so that the cluster centers will be the peaks of local density of data points. Only with this magnitude, the two methods are comparable. The matches of each cluster and the overall agreements of the experimental classification to the clustering results are listed in Table 4. As can be seen, the ALL samples (cluster A) can be better clustered by the LMC method ($M_A(\text{LMC}) = 0.792$) than by the hierarchic clustering method ($M_A(\text{HC}) = 0.784$), while the AML samples can be better described by the hierarchic clustering method ($M_B(\text{HC}) = 0.526$) than by LMC method ($M_B(\text{LMC}) = 0.521$). Overall, the experimental classification agrees better with the clustering result of the LMC method (the agreement is 0.643) than with that of the hierarchic clustering method (the agreement is 0.624).

This example shows that the LMC method, like the hierarchic clustering method, can be used for hierarchic clustering of microarray gene expression data. Unlike the hierarchic clustering method, the LMC method has the flexibility to choose magnitude properties, for example, using (6) to cluster high-differential expression genes, which will be the topic of future studies.

Table 4: Comparison of the matches and agreements of the experimental classification listed in Table 3 to the clustering results of the LMC method and the HC method.

Clusters	Matches to LMC	Matches to HC
A	0.7924	0.7836
*A*1	0.74	0.7252
*A*11	0.6304	0.6506
*A*111	0.5	0.5
*A*112	0.4358	0.4706
*A*113	0.3158	0.353
*A*12	0.6666	0.6666
*A*2	0.4444	0.4
*A*21	0.5	0.421
*A*211	0.6666	0.3076
*A*213	0.8	0.25
B	0.5208	0.5264
*B*1	0.5	0.4652
*B*11	0.0816	0.25
*B*12	0.1818	0.2858
*B*13	0.353	0.3076
*B*131	0.4	0.3636
*B*14	0.4	0.2858
*B*141	0.4444	0.3334
*B*15	0.2222	0.4
*B*2	0.1066	0.1112
*B*21	0.081	0.0846
*B*212	0.0548	0.0572
Agreement	0.643	0.624

4. CONCLUSION

This work proposed the local maximum clustering (LMC) method and evaluated its performance as compared with some typical clustering methods through designed model data sets. This clustering method is an unsupervised one and can generate hierarchic cluster structures with minimum input. It allows a magnitude property of research interest to be chosen for clustering. The comparison using model data sets indicates that the local maximum method can produce more accurate cluster results than the hierarchic, the K-mean, and the self-organized map clustering methods. As an example of application, this method is applied to cluster the leukemia samples reported in the microarray study of Golub et al. [12]. The comparison shows that the experimental classification can be better described by the cluster result from the LMC method than by the hierarchic clustering method.

REFERENCES

[1] A. Brazma and J. Vilo, "Gene expression data analysis," *FEBS Letters*, vol. 480, no. 1, pp. 17–24, 2000.

[2] M. P. Brown, W. N. Grundy, D. Lin, et al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of the National Academy of Sciences of the USA*, vol. 97, no. 1, pp. 262–267, 2000.

[3] J. K. Burgess and Hazelton R. H., "New developments in the analysis of gene expression," *Redox Report*, vol. 5, no. 2-3, pp. 63–73, 2000.

[4] J. P. Carulli, M. Artinger, P. M. Swain, et al., "High throughput analysis of differential gene expression," *Journal of Cellular Biochemistry Supplements*, vol. 30-31, pp. 286–296, 1998.

[5] J. M. Claverie, "Computational methods for the identification of differential and coordinated gene expression," *Human Molecular Genetics*, vol. 8, no. 10, pp. 1821–1832, 1999.

[6] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the USA*, vol. 95, no. 25, pp. 14863–14868, 1998.

[7] O. Ermolaeva, M. Rastogi, K. D. Pruitt, et al., "Data management and analysis for gene expression arrays," *Nature Genetics*, vol. 20, no. 1, pp. 19–23, 1998.

[8] G. Getz, E. Levine, and E. Domany, "Coupled two-way clustering analysis of gene microarray data," *Proceedings of the National Academy of Sciences of the USA*, vol. 97, no. 22, pp. 12079–12084, 2000.

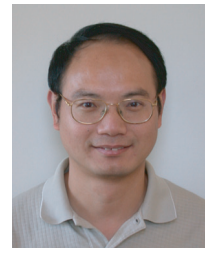
[9] P. Toronen, M. Kolehmainen, G. Wong, and E. Castren, "Analysis of gene expression data using self-organizing maps," *FEBS Letters*, vol. 451, no. 2, pp. 142–146, 1999.

[10] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nature Genetics*, vol. 22, no. 3, pp. 281–285, 1999.

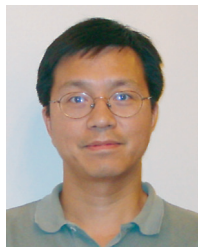
[11] P. Tamayo, D. Slonim, J. Mesirov, et al., "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences of the USA*, vol. 96, no. 6, pp. 2907–2912, 1999.

[12] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[13] M. Meila, "Comparing clusterings," UW Statistics Tech. Rep. 418, Department of Statistics, University of Washington, Seattle, Wash, USA, 2002, http://www.stat.washington.edu/mmp/#publications/.

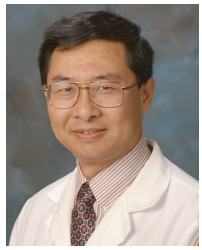


Xiongwu Wu received his B.S., M.S., and Ph.D. degrees in chemical engineering from Tsinghua University, Beijing, China. From 1993 to 1996, he was a Research Fellow in the Cleveland Clinic Foundation, Cleveland, Ohio. Then he worked as a Research Assistant Professor in George Washington University and Georgetown University. He also held an Associate Professor position in Nanjing University of Chemical Technology, Nanjing, China. Currently, Dr. Wu is a Staff Scientist at the Laboratory of Biophysical Chemistry, National Heart, Lung, and Blood Institute, National Institutes of Health, Bethesda, Maryland. His research focuses on computational chemistry and biology. His research activities include molecular simulation, protein structure prediction, electron microscopy image processing, and gene expression analysis. He has developed a series of computational methods for efficient and accurate computational studies.



Yidong Chen received his B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1983 and 1986, respectively, and his Ph.D. degree in imaging science from Rochester Institute of Technology, Rochester, NY, in 1995. From 1986 to 1988, he joined the Department of Electronic Engineering of Fudan University as an Assistant Professor. From 1988 to 1989, he was a Visiting Scholar in the Department of Computer Engineering, Rochester Institute of Technology. From 1995 to 1996, he joined Hewlett Packard Company as a Research Engineer, specialized in digital halftoning and color image processing. Currently, he is a Staff Scientist in the Cancer Genetics Branch of National Human Genome Research Institute, National Institutes of Health, Bethesda, Md, specialized in cDNA microarray bioinformatics and gene expression data analysis. His research interests include statistical data visualization, analysis and management, microarray bioinformatics, genomic signal processing, genetic network modeling, and biomedical image processing.

Bernard R. Brooks obtained his Undergraduate degree in chemistry from the Massachusetts Institute of Technology in 1976 and received his Ph.D. degree in 1979 from the University of California at Berkeley with Professor Henry F. Schaefer. His research efforts at Berkeley focused on the development of methods for electronic structure calculations. In 1980, Dr. Brooks joined Professor Martin Karplus at Harvard University as a National Science Foundation Postdoctoral Fellow where he became the primary developer of the Chemistry and Harvard Macromolecular Mechanics (CHARMM) software system, which is useful in simulating motion and evaluating energies of macromolecular systems. In 1985, Dr. Brooks joined the staff of the Division of Computer Research and Technology at the National Institutes of Health where he became the Chief of the Molecular Graphics and Simulation Section of the Laboratory of Structural Biology. Dr. Brooks is currently the Chief of the Computational Biophysics Section of the Laboratory of Biophysical Chemistry (LBC) at the National Heart, Lung, and Blood Institute (NHLBI) where he continues to develop new methods and to apply these methods to both basic and specific problems of biomedical interest.



Yan A. Su is the Associate Professor in the Department of Pathology and a member in Cardinal Bernardin Cancer Center, Loyola University Medical Center at Chicago. He received his M.D. degree in Lanzhou Medical College and Ph.D. degree in University of Michigan. He had the postdoctoral training in both of Michigan Comprehensive Cancer Center, University of Michigan, and the National Human Genome Research Institute, National Institutes of Health. Dr. Su was an Assistant Professor at Lombardi Cancer Center, Georgetown University Medical Center in 1997 and became an Associate Professor at Loyola University Chicago in 2002. His research effort focuses on molecular biology of malignant melanoma and breast cancer and he has the NIH funded projects in high-throughput analysis of gene expression. In addition, he is a member in the NIH study sections.

EURASIP Journal on Applied Signal Processing 2004:1, 64–80

Cluster Structure Inference Based on Clustering Stability with Applications to Microarray Data Analysis

Ciprian Doru Giurcăneanu
Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere, Finland
Email: cipriand@cs.tut.fi

Ioan Tăbuş
Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere, Finland
Email: tabus@cs.tut.fi

Received 28 February 2003; Revised 7 July 2003

This paper focuses on the stability-based approach for estimating the number of clusters K in microarray data. The cluster stability approach amounts to performing clustering successively over random subsets of the available data and evaluating an index which expresses the similarity of the successive partitions obtained. We present a method for automatically estimating K by starting from the distribution of the similarity index. We investigate how the selection of the hierarchical clustering (HC) method, respectively, the similarity index, influences the estimation accuracy. The paper introduces a new similarity index based on a partition distance. The performance of the new index and that of other well-known indices are experimentally evaluated by comparing the "true" data partition with the partition obtained at each level of an HC tree. A case study is conducted with a publicly available Leukemia dataset.

Keywords and phrases: clustering stability, number of clusters, hierarchical clustering methods, similarity indices, partition-distance, microarray data.

1. INTRODUCTION

The clustering algorithms are frequently used for analyzing the microarray data. While various clustering methods help the practitioner in bioinformatics to ascertain different characteristics in structural organization of microarray datasets, the task of selecting the most appropriate algorithm for solving a particular problem is nontrivial. While various clustering methods are applied in hundreds of microarray research papers, a question arises frequently, namely, how to compare two different partitions of the same dataset obtained by two different algorithms. The comparison becomes more difficult when the two partitions do not contain the same number of clusters. The accurate estimation for the number of clusters K is essential because most of the existing clustering procedures request K as input.

The robustness of the clustering algorithms is usually studied by investigating their stability with respect to perturbations changing the original dataset, for example, by drawing random subsets or by artificially adding noise [1]. The stability methods can be also used in exploratory data analysis when little prior information is available regarding the dataset, which is generally the case with microarray data. The main principle is to randomly split the dataset and cluster each subset independently, and then to check the stability (or degree of agreement) of the two obtained partitions. The clustering is stable if the cluster memberships inferred in the two subsets are similar to the memberships in the entire sample [1]. The following two different approaches have been considered when applying the stability methods for finding structure in microarray data.

(1) After randomly splitting the dataset into two subsets, select one subset for learning and another for test. Firstly, a clustering algorithm C_A is applied to the learning set, and the resulting classes are used to classify the samples which belong to the test set. Then the test set is clustered with the same algorithm C_A, and a similarity measure (index) is computed between the labels produced by classification, respectively, clustering [2, 3, 4].

(2) Apply the same clustering algorithm C_A to both subsets and calculate the similarity index on the samples belonging to the intersection of subsets [5]. A modified variant is

introduced in [6]: C_A is applied to the whole dataset (reference clustering) and to a randomly chosen subset. The similarity index is computed for the samples contained in the selected subset.

In both approaches, it is assumed that the number of clusters is $k \in \{2, 3, \ldots, k_{\max}\}$, and for each value allowed for k, after running the algorithm many times, the empirical distribution of the similarity index is collected. In [3], the number of clusters K is estimated based on the median of similarity index values. Evaluating the degree of agreement is rephrased in [4] as a prediction problem: their index ("prediction strength") $ps(k)$ measures how well the cluster centroids from the training set predict "co-memberships" in the test set. The index $ps(k)$ is averaged over several random splittings of the original data (into training set and test set), and the estimated number of clusters is given by $\hat{K} = \arg\max_{2\le k\le k_{\max}} \text{mean}[ps(k)]$ when $\max(\text{mean}[ps(k)])$ is larger than a given threshold. The approach in [6] evaluates the stability for individual patterns and clusters relying on a different similarity score called optimal association. In [5], $\hat{K}$ is chosen as the value, where there is a transition from a similarity index distribution that is concentrated near one to a wider distribution: $\hat{K}$ is visually estimated by using the empirical cumulative distribution function or, alternatively, based on the value of the 90th percentile. In consensus clustering (CC) [7], the central role is played by the consensus matrix that records, for every pair of objects, the proportion of clustering runs in which the two objects are clustered together. Based on the histogram of the consensus matrix entries, an empirical cumulative distribution function is defined, and the selection of the appropriate number of clusters proceeds by inspection of the shape of this function when $k \in \{2, 3, \ldots, k_{\max}\}$.

We propose to improve the algorithm described in [5] such that $\hat{K}$ can be automatically estimated without resorting to visual inspection or other heuristic methods. To evaluate the importance of index selection on the accuracy of the estimation, we revisit various similarity indices. Then we define and analyze a new similarity index, which is connected to the recently introduced partition distance [8]. In [3, 5], the Fowlkes-Mallows index [9] is recommended for stability-based methods, but we show experimentally that our newly introduced index and the Jaccard index [10] perform better. We also show in this paper that partition distance is useful in designing a visualization tool which helps consistently the interpretation of clustering results for microarray data.

Potentially, any clustering algorithm can be used in our settings, and we investigate the impact of the algorithm selection on the estimated $\hat{K}$. We restrict our investigation to the agglomerative hierarchical clustering (HC) algorithms [10] mainly because this class of clustering methods is very popular in microarray data analysis [11]. These algorithms are computationally efficient since the same tree can be used for all values of $k \in \{2, 3, \ldots, k_{\max}\}$ by looking at different levels of the tree each time. In [7], when evaluating the performances of CC with various microarray datasets, it was concluded that CC based on HC produces slightly better results than CC based on self-organizing maps (SOM). We remark that in [5, 6, 7] the HC is done by the group-average algorithm [10, 12]. In our simulated experiments, the group-average shows modest results when compared with complete-linkage and Ward's methods [10, 12].

The remainder of this paper is organized as follows. Section 2 includes a discussion of some results on the estimation of the number of clusters, previously reported for the publicly available Leukemia dataset [13]. In Section 3, we introduce the similarity indices. Relying on the revisited properties of the partition distance [8], a new similarity index $s(\cdot, \cdot)$ is defined, and a lower bound is found under the hypothesis of generalized hypergeometric distribution for the contingency table. In Section 4, we evaluate experimentally $s(\cdot, \cdot)$ by comparing the "true" clustering of a dataset with the partition obtained at each level of a HC tree. In Section 5, we introduce the stability-based method for finding the data structure by extending the approach proposed in [5]. Comparisons with other methods are reported for simulated data, and a case study is conducted on Leukemia dataset [13].

2. MOTIVATION OF THE WORK

In order to illustrate the challenge of structure estimation for microarray data, we consider the leukemia dataset described in [13], publicly available at http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi, which comes from a study of gene expression in two types of acute leukemias, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The true number of classes may be considered three since the biological labeling of the patient samples is ALL-B, ALL-T, and AML [13]. The dataset consists of 6817 human genes measured for 72 patients: 47 cases of ALL (38 B-cell ALL and 9 T-cell ALL) and 25 cases of AML.

We note that the clustering of Leukemia dataset was already investigated in several studies. In [13], the SOM are applied to cluster measurements from 38 patients (out of 72), relying on 50 "informative" genes selected based on a supervised procedure. We emphasize here that the "informative" genes selection relies on the gene correlation with different types of Leukemia. In two recent publications [14, 15], various validation techniques based on computing internal indices are used to estimate the number of clusters in the 38×50 dataset when SOM is the clustering algorithm. The paper [15] concludes that the estimated number of clusters is $\hat{K} = 2$ and mentions, as a second best choice, $\hat{K} = 4$.

The whole set of measurements from the 72 patients is clustered in [16] by k-means, fuzzy c-means networks, SOM, fuzzy SOM, and growing cell structure (GCS) algorithm. When varying the number of clusters between 2 and 16, all the resulting clusterings are evaluated based on the distribution of Leukemia types within the clusters, the highest degree of intracluster homogeneity being obtained when samples are divided into 9 clusters by fuzzy SOM. A procedure for gene selection is applied.

In [3], the 72 tumors from Leukemia dataset are clustered by partitioning around medoids (PAM) [10] after selecting 100 genes which have the largest variance across tumor

samples: for $\hat{K} = 3$, one ALL B-cell sample is clustered with the ALL T-cell samples, and the rest of the observations are allocated correctly. Results on estimating the number of clusters are also reported: applying *clest*, *kl* [17], *hart* [18], or silhoutte (*sil*) [10] leads to $\hat{K} = 3$; *ch* [19] estimates $\hat{K} = 2$. The estimated number of clusters is $\hat{K} = 10$ when using *gap* [20], and $\hat{K} = 5$ when employing *gapPC* [20]. Note that *clest* was originally introduced in [3] and extends the stability-based approach from [2]. Another method relying on stability principle, CC, was formalized and tested in [7]. Since their settings allow to apply various clustering methods, results on estimated number of clusters for 38 (out of 72) samples of Leukemia dataset are reported when using HC and SOM. The method CC in conjunction with HC leads to $\hat{K} = 5$, and to $\hat{K} = 4$ when employing CC in combination with SOM.

In light of these results reported for the Leukemia dataset, we can better understand the importance and difficulty of validation of the number of clusters. It becomes apparent that every method for structure estimation must be deeply analyzed and validated with simulated data for which the true nature is known before applying it to analyze the microarray data. Leukemia dataset is also a good example for illustrating the paradigm of "high dimension and small sample size" which is common in microarray data analysis. It was pointed out in [7] that this paradigm prevents the use of some clustering algorithms, and we show in this paper how stability methods can circumvent this difficulty.

3. SIMILARITY MEASURES

Given an N-object set $T = \{O_1, O_2, \ldots, O_N\}$, suppose that $P = \{P_1, P_2, \ldots, P_r\}$ and $P' = \{P'_1, P'_2, \ldots, P'_c\}$ represent two distinct partitions of T, that is, $\bigcup_{i=1}^{r} P_i = \bigcup_{i=1}^{c} P'_i = T$, where $P_i \bigcap P_j = \emptyset$ for $1 \le i \ne j \le r$ and $P'_i \bigcap P'_j = \emptyset$ for $1 \le i \ne j \le c$. We name, in the sequel, any nonempty subset of T *cluster*. So, any partition of T is a set of mutually exclusive clusters whose reunion is T.

The partitions P and P' are identical if and only if every cluster in P is a cluster in P'. Let M be an $r \times c$ matrix where the quantity m_{ij} is the number of objects in common between the ith cluster of P and the jth cluster of P'. The contingency table is represented in Table 1, where $m_{i\cdot} \triangleq \sum_{j=1}^{c} m_{ij}$ for $1 \le i \le r$ and $m_{\cdot j} \triangleq \sum_{i=1}^{r} m_{ij}$ for $1 \le j \le c$. It is easy to observe that $m_{\cdot\cdot} \triangleq \sum_{i=1}^{r} m_{i\cdot} = \sum_{j=1}^{c} m_{\cdot j} = N$.

3.1. Rand, Jaccard, and Fowlkes-Mallows similarity indices

We introduce the following function relative to an arbitrary partition P of T: for any pair of distinct objects $(O_\ell, O_m) \in T^2$, $1 \le \ell < m \le N$,

$$\mathbf{1}_P(O_\ell, O_m) \triangleq \begin{cases} 1, & \exists i \in \{1, 2, \ldots, |P|\} \text{ such that } \{O_\ell, O_m\} \subseteq P_i, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

Table 1: The contingency table for the partitions P and P' of the N-object set T.

	Cluster	Partition P'				Sums
		P'_1	P'_2	$\cdots$	P'_c	
Partition P	P_1	m_{11}	m_{12}	$\cdots$	m_{1c}	$m_{1\cdot}$
	P_2	m_{21}	m_{22}	$\cdots$	m_{2c}	$m_{2\cdot}$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	P_r	m_{r1}	m_{r2}	$\cdots$	m_{rc}	$m_{r\cdot}$
	Sums	$m_{\cdot 1}$	$m_{\cdot 2}$	$\cdots$	$m_{\cdot c}$	$m_{\cdot\cdot} = N$

which indicates if two objects belong to the same cluster in the partition P.

Following a classic procedure, we firstly define four sets:

$$\begin{aligned} \mathcal{W}_1 &\triangleq \{(O_\ell, O_m) \in T^2 \mid \mathbf{1}_P(O_\ell, O_m) = 1,\ \mathbf{1}_{P'}(O_\ell, O_m) = 1\}, \\ \mathcal{W}_2 &\triangleq \{(O_\ell, O_m) \in T^2 \mid \mathbf{1}_P(O_\ell, O_m) = 1,\ \mathbf{1}_{P'}(O_\ell, O_m) = 0\}, \\ \mathcal{W}_3 &\triangleq \{(O_\ell, O_m) \in T^2 \mid \mathbf{1}_P(O_\ell, O_m) = 0,\ \mathbf{1}_{P'}(O_\ell, O_m) = 1\}, \\ \mathcal{W}_4 &\triangleq \{(O_\ell, O_m) \in T^2 \mid \mathbf{1}_P(O_\ell, O_m) = 0,\ \mathbf{1}_{P'}(O_\ell, O_m) = 0\}, \end{aligned} \tag{2}$$

and denote the cardinalities of these sets, $w_i \triangleq |\mathcal{W}_i|$ for $i \in \{1, 2, 3, 4\}$. Then we recall the definitions for three well-known similarity indices:

(1) Rand [21]: $(w_1 + w_4)/\sum_{i=1}^{4} w_i$,
(2) Jaccard [22]: $w_1/\sum_{i=1}^{3} w_i$,
(3) Fowlkes-Mallows [9]: $w_1/\sqrt{(w_1 + w_2)(w_1 + w_3)}$.

Since w_i $(1 \le i \le 4)$ are nonnegative numbers, all three indices take values in the interval $[0, 1]$. The partitions P and P' are identical if and only if $w_2 = w_3 = 0$; when they are identical and $w_1 \ne 0$, then all indices are equal to their maximum value 1. Observe for the denominator of Rand index that $\sum_{i=1}^{4} w_i = \binom{N}{2}$. The Jaccard index is not defined for the trivial case when each cluster in P and P' contains at most 1 object, which is equivalent to $w_1 = w_2 = w_3 = 0$. The Fowlkes-Mallows index is not defined when $w_1 = w_2 = 0$ (each cluster in P contains at most 1 object) or $w_1 = w_3 = 0$ (each cluster in P' contains at most 1 object). Formulae for fast computing w_i $(1 \le i \le 4)$ are available [23].

To each similarity measure $sm(P, P')$, bounded by zero and unity, we can associate a dissimilarity $d(P, P') \triangleq 1 - sm(P, P')$; in some cases, $d(P, P')$ could be a metric on the set of all partitions of a given set of objects T [12]. In the next section, we start from the definition given in [8] for the partition distance (which is a metric) and define a new similarity index.

3.2. A similarity index defined as complement of a partition distance

In [8], the following definition is introduced for the partition distance $D(P, P')$ between P and P': "$D(P, P')$ is the minimum number of elements that must be deleted from

T, so that the two induced partitions (P and P' restricted to the remaining elements) are identical." It was pointed out in [24] that the partition distance is also equal to the minimum number of elements that must be moved between clusters in P, so that the resulting partition equals P' (with the convention that any set which becomes empty is no longer a cluster).

Proposition 1. *The partition distance $D(P, P')$ is a metric on the set of all partitions of a given set of objects T.*

Proof. See Appendix A. □

An *assignment* is a selection of entries of the contingency matrix M such that no row or column contains more than one selected entry and is called *optimal* when the sum of the selected cell values is the largest over all possible assignments [24]. Let $A(P, P')$ denote the value of the optimal assignment for the contingency matrix M.

Theorem 1 [24]. *Two properties of partition distance:*

(a) *The relationship between the partition distance and the optimal assignment is given by $D(P, P') = N - A(P, P')$.*

(b) *The elements to be removed from T to induce identical partitions on P and P', are all those objects not associated with any selected cells of the optimal assignment.*

The proof of the theorem is given in [24] where the theorem is further used to show how the partition distance can be computed in $\mathbb{O}((r + c)^3)$ time after creating the matrix M in $\mathbb{O}(N)$ time. Note that the initial algorithm proposed in [8] to compute $D(P, P')$ for any pair of partitions (P, P') is an exponential-time algorithm, and the algorithm in [24] reduces dramatically the computational complexity.

Proposition 2. *The maximum of the partition distance is $\max_{(P,P')} D(P, P') = N - 1$ and is achieved if and only if one partition consists of a single cluster and the other one consists only of clusters containing single-objects.*

Proof. See Appendix A. □

The above results suggest the definition of the following index of similarity between any two partitions P and P':

$$s(P, P') \triangleq 1 - \frac{D(P, P')}{N - 1} = \frac{A(P, P') - 1}{N - 1}. \tag{3}$$

The new index is a measure of similarity ranging from $s(P, P') = 0$ when the two partitions have no similarities (i.e., when one consists of a single cluster and the other only of clusters containing single-objects) to $s(P, P') = 1$ when the partitions are identical.

Any injective mapping $\sigma : \{1, 2, \ldots, |P'|\} \to \{1, 2, \ldots, |P|\}$ ($|P'| \leq |P|$) is called association [6] and is useful for comparing two partitions P and P' defined over an N object set T. The measure of similarity between P and P' is computed as $s^*(P, P') \triangleq \max_{\sigma(\cdot)}(1/N) \sum_{j=1}^{|P'|} m_{\sigma(j),j}$ where $m_{\cdot,\cdot}$ denotes the entries of the contingency matrix. Observe that $s^*(P, P') = A(P, P')/N$ and is close to the similarity index defined in (3); $A(P, P') \leq N$ implies that $s(P, P') \leq s^*(P, P')$. It is noticed in [6] that the computation of $s^*(P, P')$ by brute-force enumeration is exponential in the number of clusters, and therefore an approximative greedy heuristic was used there for finding a suboptimal association $\sigma(\cdot)$. Since then, the fast algorithm was introduced in [24], and hence we are going to use the fast, nonapproximative evaluation of $s(P, P')$.

Table 2: The contingency table for the Leukemia dataset: the true partition given by a priori knowledge on the type of disease for each patient is compared with the partition produced by complete-linkage algorithm when $\hat{K} = 3$. All the 3571 genes are used for clustering. The entries associated to the optimal assignment are represented in bold.

Cluster	ALL B-cell	ALL T-cell	AML
C_1	**26**	8	8
C_2	7	**0**	2
C_3	5	1	**15**

We observe that the definition of both $s(P, P')$ and $s^*(P, P')$ relies on the optimal assignment $A(P, P')$, and the main difference between these similarity indices is given by the normalization procedure. Since in [6] $s^*(P, P')$ was successfully applied for detecting stable clusters in microarray data, we are encouraged to employ $s(P, P')$ in stability-based methods for analyzing data produced by microarray technology. The superiority of our approach consists in using nonapproximative algorithms for computing the similarity index.

The use of $s(\cdot, \cdot)$ in validation of microarray data clustering is appealing since the optimal assignment lends itself to be employed as a visualization tool. Assume that we depict the contingency matrix defined by two partitions P and P', where P corresponds to the classes in a microarray dataset already known from medical evidence while P' contains classes found for the same dataset after running a clustering algorithm. Representing in bold the entries associated to the optimal assignment will allow the investigator to assess very easily the memberships. The procedure does not require the number of clusters to be the same in the compared partitions. Moreover, the number of clusters can be visually assessed by checking that all entries in the optimal assignment are larger than zero. Examples of such representations are given in Section 5.2, Tables 2 and 8. When the true state of the nature is not known, the same graphical representation can be used for comparing the results of two different clustering algorithms.

3.3. Similarity indices "corrected for chance"

A similarity index is "corrected for chance" when the expectation of the index takes some constant value (e.g., zero) under an appropriate null model for the contingency table. The property is discussed in [25], and the following general formula is proposed to correct an index:

$$\frac{\text{Index} - \text{Expected Index}}{\text{Maximum Index} - \text{Expected Index}}. \tag{4}$$

The most popular null model assumes that the $r \times c$ contingency table ($r \geq c$) is constructed from the generalized hypergeometric distribution. The main hypothesis is that the two partitions are mutually independent and subject to the condition that the cluster sizes are fixed at $(\alpha_1, \alpha_2, \ldots, \alpha_r)$ and $(\beta_1, \beta_2, \ldots, \beta_c)$, respectively. The α_i and β_j are the marginal totals of m_{ij}, namely, $m_{i\cdot}$ and $m_{\cdot j}$, respectively. Then the expectation of m_{ij} is $E[m_{ij}] = \alpha_i\beta_j/N$ [9, 25]. For example, correcting the Rand index under this hypothesis leads to the expression

$$\text{Adjusted Rand} = \frac{w_1 + w_4 - N_c}{\sum_{i=1}^{4} w_i - N_c}, \tag{5}$$

where two different formulae were proposed for N_c in [25, 26]. We use in the sequel the notations Rand_{HA} and Rand_{MA} for the adjusted index defined in [25], respectively, [26].

We investigate in Appendix B the existence of a lower bound for the expectation of the similarity index defined by (3) when the hypothesis of generalized hypergeometric distribution is verified.

It was already pointed out in [3] that the assumption on the statistical independence of the two compared clusterings does not hold for stability methods since the same data are used to produce both partitions. To gain more insights on the possibility of using $s(\cdot,\cdot)$ in practical applications, we study in Appendix C the asymptotic and finite characteristics of $s(\cdot,\cdot)$ and compare them with the characteristics of other similarity indices.

4. USING THE SIMILARITY INDEX $s(\cdot,\cdot)$ IN HIERARCHICAL CLUSTER ANALYSIS

The aim of this section is to evaluate experimentally $s(\cdot,\cdot)$ when we assume that the "true" structure of the data (the number of clusters and the membership) is known and compare this partition with the partition obtained at each level of a HC tree.

It is a well-known fact that the HC does not yield a discrete number of clusters, but rather a hierarchical arrangement between objects. For better understanding of the behavior of similarity indices, assume that the "true" structure of the data is known and compare this partition with the partition obtained at each level of the HC solution. This approach was originally used in [27] to compare Rand, Rand_{HA}, Rand_{MA}, Fowlkes-Mallows, and Jaccard indices.

We reconsider the experiments described in [27] to evaluate the newly introduced index $s(\cdot,\cdot)$, and for comparison, we compute also Rand, Rand_{HA}, and Jaccard indices. For the first set of experiments, each generated dataset consists of 50 points uniformly distributed in a hypercube in 4-, 6-, or 8-dimensional Euclidean space. There is no significant cluster structure in the data, but a "criterion" solution is assumed: a hypothetical number of clusters (set at either 2, 3, 4, or 5) and a particular distribution pattern of the points to the clusters. Three density patterns are used: equal density (objects are uniformly assigned across the clusters), 10% density condition (one cluster contains 10% of the total number of objects, while 90% of objects are uniformly assigned across the other clusters), and 60% density condition (one cluster contains 60% of the total number of objects, while 40% of objects are uniformly assigned across the other clusters). For example, when the number of clusters is 5 for 10% density, the points are assigned to the clusters as follows: 5, 11, 11, 11, 12. For each selected number of clusters and for each pattern distribution, 15 datasets are generated. The HC is performed by using the single link, the complete link, the group average, and the Ward method [12]. The computed similarity index is averaged over the datasets and over the HC methods, and the mean statistics (with limits at two standard deviation) are plotted in Figures 1a, 1b, and 1c versus the hierarchy level for each of the three density conditions. The two-standard deviation limit is omitted for those levels where the values would be negative or larger than 1.0. The only index for which the mean plot is flat and close to zero is Rand_{HA}. For $s(\cdot,\cdot)$ and Jaccard, the computed mean is decreasing when the number of clusters in HC is increasing. Rand takes values larger than the other indices, and the mean is increasing slowly when the number of clusters in HC is increasing. For $s(\cdot,\cdot)$ and Jaccard, the variance is larger when the partition contains a small cluster; in the same situation, we observe a serious increase in the variance of Rand.

In the second set of experiments, the test data are generated according to the algorithm described in [28]; the clusters contained in the data are separated in the variable space and are internally cohesive. It was observed that the mean of similarity indices is close to 1.0 when the number of clusters in HC solution is equal to the true number of clusters for all considered structures. We plot in Figure 1d the mean statistics for the similarity indices in the case of 60% density condition for four clusters.

All plots in Figure 1 for Rand, Rand_{HA}, and Jaccard are very close to similar plots in [27]. The new index $s(\cdot,\cdot)$ has almost the same performance pattern as Jaccard; generally, the variance of $s(\cdot,\cdot)$ is smaller than the variance of Jaccard index, while the mean is larger. Extending the conclusions from [27], we can observe that a value larger than 0.9 for the Rand, 0.7 for the Jaccard, and 0.8 for $s(\cdot,\cdot)$ is likely to reflect the recovery of some part of the true structure.

For all structured datasets, the clusters contained in the data have been crafted to be disjointed, separated in the variable space, and internally cohesive. Relying on these properties to obtain grouping in k clusters ($2 \leq k \leq k_{\max}$), we choose the clusters at kth depth in the dendrogram. In microarray cluster analysis, the datasets contain outliers which do not belong to any group. Consequently, the dendrogram resulting after running a certain HC algorithm could have at kth depth a singleton (a cluster containing only an outlier). In that case, we move down the HC tree until k distinct clusters are identified, each of them containing at least two objects. It was shown in [29] that the similarity with the true partition is larger when considering the k distinct clusters (and ignoring the outliers) than simply taking all clusters at kth depth in the dendrogram. Since we aim to identify structures in data, we prefer an algorithm which can accurately

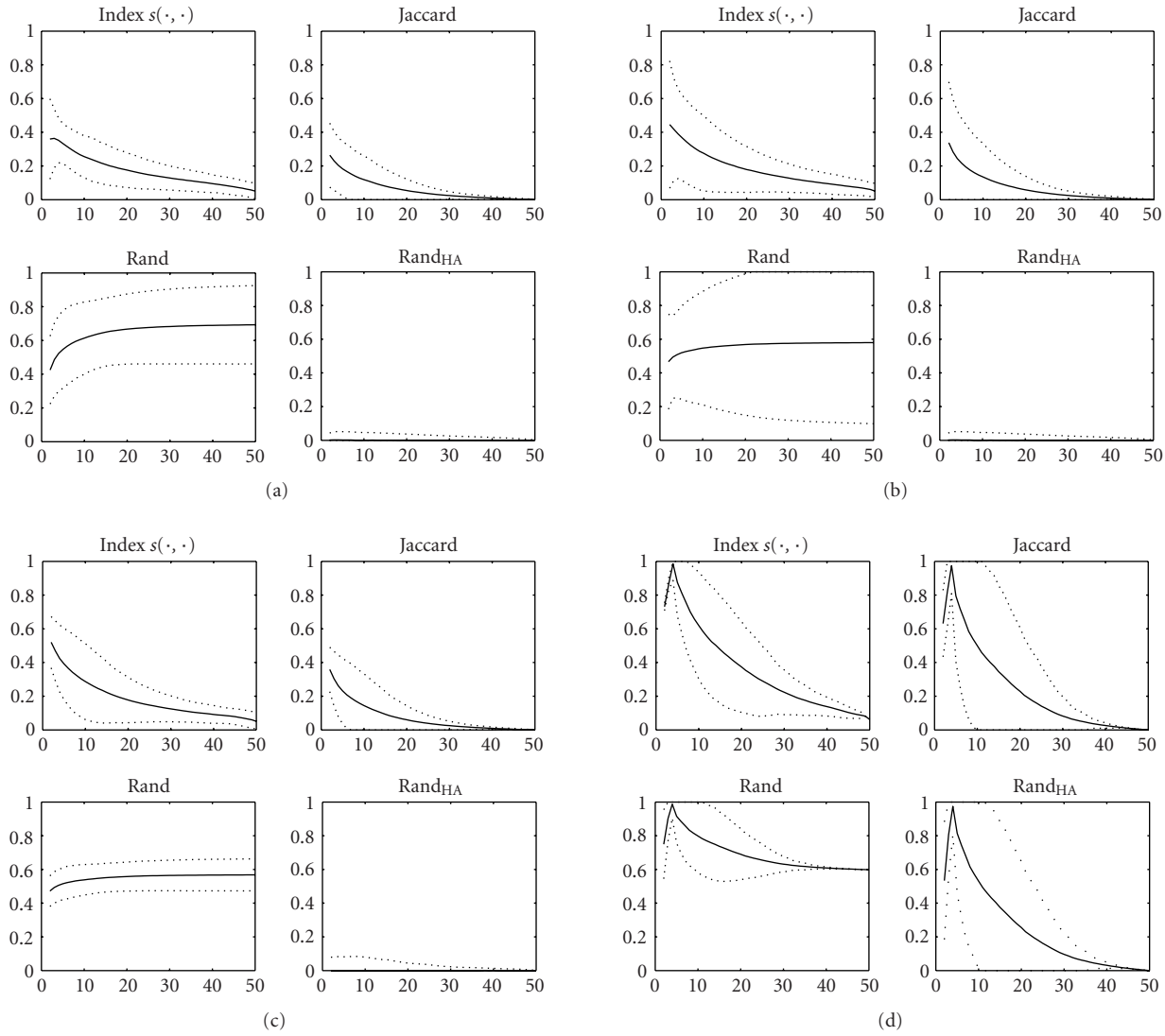


FIGURE 1: Mean of the similarity indices versus the number of clusters (solid line) with limits at two-standard deviation (dotted line). (a) The equal density condition when no structure exists in the data. (b) The 10% density condition when no structure exists in the data. (c) The 60% density condition when no structure exists in the data. (d) The 60% density condition when data contains four distinct clusters The x-axes denote the number of clusters while the y-axes denote the similarity index value.

estimate $\hat{K}$ relying on a subset of the original dataset instead of one that clusters all objects with an increased risk of misclassification.

5. STABILITY-BASED METHOD FOR ESTIMATING THE NUMBER OF CLUSTERS

First, we briefly revisit the algorithm introduced in [5] when the dataset contains N points embedded in p-dimensional space. Assume that the maximum number of clusters is $k_{\max}$, and for each allowable value of k, except the trivial case ($k = 1$), select from the data two subsets such that each of them contains $f = 80\%$ of the original samples. Use the average-link HC algorithm [12] to cluster every subset in k nonsingleton groups, and then compute the Fowlkes-Mallows similarity index [9] on the intersection of subsets. The number of pairs of solutions compared for each k is $N_t = 100$. It was pointed out in [5] that the histogram of similarity indices is concentrated near one only for values of k smaller than or equal to the "true" number of clusters. Relying on this observation, the number of clusters has been visually evaluated by inspecting the plot of the empirical cumulative distribution function of similarity index. We extend the algorithm from [5] for any similarity index and any HC algorithm.

In the rest of the section, we introduce and analyze the method for automatic selection of the number of clusters. Let $sm_{k,t}$ be the value of the similarity index for the tth pair of solutions compared under the hypothesis of k nonsingleton

clusters. For a given k, we are interested in the histogram obtained from the values $sm_{k,1}, sm_{k,2}, \ldots, sm_{k,N_t}$. A good indicator for the location of the histogram is the *mean* of the values, but since the *median* is more robust to the presence of outliers, we compute

$$m_k \triangleq \text{median}\,(sm_{k,1}, sm_{k,2}, \ldots, sm_{k,N_t}). \tag{6}$$

We decide that there is no significant structure in the analyzed data ($\hat{K} = 1$) if

$$\max_{2\le k\le k_{\max}} m_k < Th, \tag{7}$$

where Th depends on the similarity index and the HC algorithm. The threshold Th is determined under a suitable null hypothesis: the *uniformity hypothesis* states that the data are sampled from a uniform distribution in p-dimensional space, while under the *unimodality hypothesis*, the data are thought to be random sample from a multivariate normal distribution [3]. We use in the sequel the uniformity hypothesis for the null case.

When $\max_{2\le k\le k_{\max}} m_k \ge Th$, let $\nu : \{2, 3, \ldots, k_{\max}\} \to \{2, 3, \ldots, k_{\max}\}$ be a permutation such that $m_{\nu(2)}, m_{\nu(3)}, \ldots, m_{\nu(k_{\max})}$ are the elements of the set $\{m_2, m_3, \ldots, m_{k_{\max}}\}$, decreasingly ordered. Calculate

$$i^* \triangleq \arg\max_i \left(m_{\nu(i)} - m_{\nu(i+1)}\right), \tag{8}$$

which is a "border" between values of k yielding stable, respectively, unstable clustering. The estimated number of clusters is given by the maximum value of k for which the resulting clustering is still stable, or equivalently $\hat{K} = \max(\nu(2), \nu(3), \ldots, \nu(i^*))$.

The improvement proposed for the algorithm described in [5] leads to an automatic procedure for estimating $\hat{K}$ without resorting to any heuristic method. The accuracy of the new algorithm is tested next using artificial and microarray data.

5.1. Performance evaluation with simulated data

We investigate the performances of the algorithm by using artificially generated data for which the true state of the nature is known. The experiments are intended for studying the influence of the HC algorithm and the similarity index on the accuracy of estimation. In [3, 5], the use of Fowlkes-Mallows similarity index is recommended. Due to this reason, we report estimation results when applying it in conjunction with group-average, complete-linkage, Ward's method, centroid, and single-linkage clustering, while for other considered indices, the comparisons are restricted to three clustering algorithms. A complete description of the clustering algorithms could be found in [10, 12]. In all cases, the distance between two clustered objects is taken to be the Euclidean distance.

The artificial data are generated according to Models 1–8 introduced in [3]: Model 1 obeys the uniformity hypothesis and Models 2–8 assume the presence of various number of clusters. For each model, $N_d = 50$ datasets are simulated, and the results are reported in Tables 3, 4, and 5, where $k_{\max} = 7$ is assumed. In Tables 3, 4, and 5, the maximum of the distribution for $\hat{K}$ over $N_d = 50$ estimations is represented in bold for each method. For every dataset, the number of pairs of solutions compared for each k ($2 \le k \le k_{\max}$) is $N_t = 100$. We note that for Models 1–8, the number of samples in every dataset varies between 100 and 200 [3] and during the subsampling process we select from the data two subsets such that each of them contains $f = 80\%$ of original samples.

For each model, the best solution corresponds to the method having the highest percentage of simulations for which the number of clusters is correctly recovered and is marked with an arrow ($\Leftarrow$) in Tables 3, 4, and 5. The only clustering algorithms that lead to good results are complete-linkage and Ward's method; the former gives 4 and the latter 8 "best solutions." The group-average clustering is recommended in [5, 6], but we remark the modest performances of the algorithm for the actual tests. Only one similarity index "corrected for chance" is considered in these experiments, namely, Rand_{HA}. Unsurprisingly, Rand_{HA} distinguishes very well between structured and unstructured datasets; when applied in conjunction with complete-linkage or Ward's method, it identifies the lack of structure for all files generated according to Model 1 ($K = 1$) and for the files associated to Models 2–8, the estimated $\hat{K}$ is always larger than 1. When the HC is based on group-average and the similarity index is Rand_{HA}, five false positive results are reported ($\hat{K} > 1$ five times for Model 1), respectively, five false negative results ($\hat{K} = 1$ five times for Model 7). The values of the threshold Th used in (7) to decide for the Models 1–8 if there is no significant structure in the analyzed dataset ($\hat{K} = 1$) are given in Table 6.

For structured Models 2–8, the best solution is associated only once to the algorithm which measures the similarity with Rand_{HA}, and this occurs for Model 5 (Table 4). Comparing the performances of various similarity indices over all models, we observe that $s(\cdot,\cdot)$ leads to the best solution five times (Models 1, 2, 3, 6, 8), Jaccard three times (Models 1, 4, 7), Rand_{HA} three times (Models 1, 5), while Fowlkes-Mallows only once (Model 1). We remark that the newly introduced index $s(\cdot,\cdot)$ is best ranked. When clustering is done by group-average, measuring the similarity with Fowlkes-Mallows index leads to poor results.

We dub *sw*, the stability-based method, for estimating the number of clusters when Ward's algorithm is used in conjunction with $s(\cdot,\cdot)$ and compare it, for the Models 1–8, with seven methods analyzed in [3]: prediction-based resampling *clest*, *gap* and *gapPC* [20], *sil* [10], *ch* [19], *kl* [17], and *hart* [18]. A description for all seven methods can be found in [3]. The bar plots in Figure 2 represent the percentage of simulations for which the number of clusters was correctly estimated by each considered method according to Tables 3, 4, and 5, respectively [3, Table 3]. By their design, *sil*, *ch*, and *kl* cannot detect the lack of structure, so for these methods, $\hat{K} \ge 2$. The plots in Figure 2 show that excepting *sw* and *clest*, all methods fail to estimate the number of clusters for at least

Table 3: Estimated number of clusters in simulated data. Results for the Models 1, 2, 3.

Similarity index	Hierarchical clustering method	Number of clusters							
Model 1 (1 cluster in 10 dimensions)									
		1*	2	3	4	5	6	7	
	Group-average	**21**	12	16	1	0	0	0	
$s(\cdot,\cdot)$	Complete-linkage	**44**	6	0	0	0	0	0	
	Ward's method	**50**	0	0	0	0	0	0	⇐
	Group-average	16	**23**	10	1	0	0	0	
Jaccard	Complete-linkage	**45**	5	0	0	0	0	0	
	Ward's method	**50**	0	0	0	0	0	0	⇐
	Group-average	15	**17**	16	2	0	0	0	
	Complete-linkage	**44**	6	0	0	0	0	0	
Fowlkes-Mallows	Ward's method	**50**	0	0	0	0	0	0	⇐
	Centroid method	0	**14**	7	7	11	11	0	
	Single-linkage	**19**	7	2	9	5	8	0	
	Group-average	**45**	4	1	0	0	0	0	
Rand_{HA}	Complete-linkage	**50**	0	0	0	0	0	0	⇐
	Ward's method	**50**	0	0	0	0	0	0	⇐
Model 2 (3 clusters in 2 dimensions)									
		1	2	3*	4	5	6	7	
	Group-average	0	1	13	**21**	14	0	1	
$s(\cdot,\cdot)$	Complete-linkage	0	0	**38**	10	2	0	0	⇐
	Ward's method	0	0	**25**	19	5	1	0	
	Group-average	0	1	13	**20**	13	2	1	
Jaccard	Complete-linkage	0	0	**35**	9	6	0	0	
	Ward's method	0	2	**35**	11	1	1	0	
	Group-average	0	1	11	**20**	15	2	1	
	Complete-linkage	0	0	**31**	10	7	1	1	
Fowlkes-Mallows	Ward's method	0	1	**34**	13	1	1	0	
	Centroid method	0	0	12	14	**15**	9	0	
	Single-linkage	3	4	**14**	9	7	5	8	
	Group-average	0	0	15	**20**	13	1	1	
Rand_{HA}	Complete-linkage	0	0	**34**	9	5	0	2	
	Ward's method	0	1	**35**	12	1	1	0	
Model 3 (4 clusters in 10 dimensions, 7 noise variables)									
		1	2	3	4*	5	6	7	
	Group-average	0	2	7	**17**	10	14	0	
$s(\cdot,\cdot)$	Complete-linkage	0	1	10	**21**	12	6	0	
	Ward's method	0	1	4	**39**	5	1	0	⇐
	Group-average	0	4	13	**15**	9	9	0	
Jaccard	Complete-linkage	0	1	14	**15**	10	10	0	
	Ward's method	0	2	9	**35**	4	0	0	
	Group-average	0	4	**13**	12	10	11	0	
	Complete-linkage	0	1	12	**13**	12	12	0	
Fowlkes-Mallows	Ward's method	0	2	7	**33**	6	2	0	
	Centroid method	0	3	10	11	**14**	10	2	
	Single-linkage	0	4	4	10	**14**	8	10	
	Group-average	0	4	13	**16**	9	8	0	
Rand_{HA}	Complete-linkage	0	1	12	**14**	11	12	0	
	Ward's method	0	2	9	**30**	5	2	2	

Table 4: Estimated number of clusters in simulated data. Results for the Models 4, 5, 6.

Similarity index	Hierarchical clustering method	Number of clusters							
Model 4 (4 clusters in 10 dimensions)									
		1	2	3	4*	5	6	7	
	Group-average	0	0	1	**23**	12	12	2	
$s(\cdot,\cdot)$	Complete-linkage	0	0	0	**34**	12	4	0	
	Ward's method	0	0	0	**36**	14	0	0	
	Group-average	0	2	4	**20**	8	12	4	
Jaccard	Complete-linkage	0	0	3	**24**	15	7	1	
	Ward's method	0	0	0	**41**	8	1	0	⇐
	Group-average	0	2	4	**18**	8	14	4	
	Complete-linkage	0	0	3	10	**20**	14	3	
Fowlkes-Mallows	Ward's method	0	0	0	**31**	16	2	1	
	Centroid method	0	2	2	**19**	14	7	6	
	Single-linkage	2	0	3	**14**	11	6	**14**	
	Group-average	0	2	4	**17**	10	13	4	
Rand_{HA}	Complete-linkage	0	0	3	14	**16**	10	7	
	Ward's method	0	1	0	**33**	12	2	2	
Model 5 (2 elongated clusters in 3 dimensions)									
		1	2*	3	4	5	6	7	
	Group-average	0	**17**	5	6	2	7	13	
$s(\cdot,\cdot)$	Complete-linkage	0	**26**	10	11	0	1	2	
	Ward's method	0	**17**	4	7	5	7	10	
	Group-average	0	**24**	15	3	2	4	2	
Jaccard	Complete-linkage	0	**27**	21	2	0	0	0	
	Ward's method	0	**26**	14	4	3	1	2	
	Group-average	0	**21**	12	6	2	6	3	
	Complete-linkage	0	22	**26**	2	0	0	0	
Fowlkes-Mallows	Ward's method	0	**21**	16	5	4	2	2	
	Centroid method	0	**20**	13	6	3	5	3	
	Single-linkage	0	10	**15**	10	7	7	1	
	Group-average	0	**20**	13	2	2	3	10	
Rand_{HA}	Complete-linkage	0	**33**	15	2	0	0	0	⇐
	Ward's method	0	**25**	14	3	2	1	5	
Model 6 (2 elongated clusters in 10 dimensions, 7 noise variables)									
		1	2*	3	4	5	6	7	
	Group-average	1	**12**	10	7	5	4	11	
$s(\cdot,\cdot)$	Complete-linkage	3	**47**	0	0	0	0	0	⇐
	Ward's method	0	**42**	6	2	0	0	0	
	Group-average	0	**14**	7	9	5	3	12	
Jaccard	Complete-linkage	4	**46**	0	0	0	0	0	
	Ward's method	0	**42**	6	1	1	0	0	
	Group-average	0	**12**	7	9	6	4	**12**	
	Complete-linkage	4	**45**	1	0	0	0	0	
Fowlkes-Mallows	Ward's method	0	**39**	9	1	1	0	0	
	Centroid method	0	**14**	9	6	10	11	0	
	Single-linkage	3	12	7	1	5	**15**	7	
	Group-average	0	1	0	0	1	1	**47**	
Rand_{HA}	Complete-linkage	0	0	0	0	0	0	**50**	
	Ward's method	0	**35**	7	1	0	0	7	

Table 5: Estimated number of clusters in simulated data. Results for the Models 7 and 8.

Similarity index	Hierarchical clustering method	Number of clusters							
Model 7 (2 overlapping clusters in 10 dimensions, 9 noise variables)									
		1	2*	3	4	5	6	7	
	Group-average	**15**	13	**15**	7	0	0	0	
$s(\cdot,\cdot)$	Complete-linkage	2	**43**	5	0	0	0	0	
	Ward's method	0	**47**	3	0	0	0	0	
	Group-average	14	13	**18**	5	0	0	0	
Jaccard	Complete-linkage	2	**46**	2	0	0	0	0	
	Ward's method	0	**48**	2	0	0	0	0	⇐
	Group-average	13	12	**17**	7	1	0	0	
	Complete-linkage	2	**46**	2	0	0	0	0	
Fowlkes-Mallows	Ward's method	0	**47**	3	0	0	0	0	
	Centroid method	6	**16**	8	6	5	4	5	
	Single-linkage	**19**	13	0	1	1	5	11	
	Group-average	5	5	3	1	0	0	**36**	
Rand_{HA}	Complete-linkage	0	5	3	0	1	0	**41**	
	Ward's method	0	**32**	3	0	0	0	15	
Model 8 (3 overlapping clusters in 13 dimensions, 10 noise variables)									
		1	2	3*	4	5	6	7	
	Group-average	10	6	7	3	2	0	**22**	
$s(\cdot,\cdot)$	Complete-linkage	0	**39**	10	1	0	0	0	
	Ward's method	0	0	**38**	11	1	0	0	⇐
	Group-average	**21**	10	7	3	1	0	8	
Jaccard	Complete-linkage	0	**37**	6	1	5	1	0	
	Ward's method	0	10	**31**	8	1	0	0	
	Group-average	**19**	10	8	3	1	0	9	
	Complete-linkage	0	**35**	6	2	6	1	0	
Fowlkes-Mallows	Ward's method	0	7	**27**	13	3	0	0	
	Centroid method	**38**	4	1	1	2	2	2	
	Single-linkage	4	7	3	1	1	0	**34**	
	Group-average	0	7	7	2	1	0	**33**	
Rand_{HA}	Complete-linkage	0	**16**	6	9	4	6	9	
	Ward's method	0	6	**25**	10	3	0	6	

Table 6: The threshold Th used in (7) to decide for the Models 1–8 if there is no significant structure in the analyzed dataset ($\hat{K} = 1$). Remark that the value of Th depends on the HC algorithm and the similarity index.

	$s(\cdot,\cdot)$	Jaccard	Fowlkes-Mallows	Rand_{HA}
Group-average	0.9350	0.8600	0.9220	0.3750
Complete-linkage	0.6260	0.4285	0.6040	0.1938
Ward's method	0.7234	0.4532	0.6255	0.3156
Centroid	—	—	0.9620	—
Single-linkage	—	—	0.9750	—

one model: *gap* for Models 5 and 6, *gapPC* for Model 6, *sil* for Model 8, *ch* for Models 5 and 8, while *hart* for Models 1, 2, 5, and 6. Since *hart* fails in four models out of eight, it is concluded in [3] that it performs the worst; *kl* does not really fail in any model, but the results are poor for Models 6–8. In all these cases the percentage of correct estimation is lower than 40%. The methods *sw* and *clest* prove to be robust; the worst result of *sw* occurs in Model 5, while the worst result of

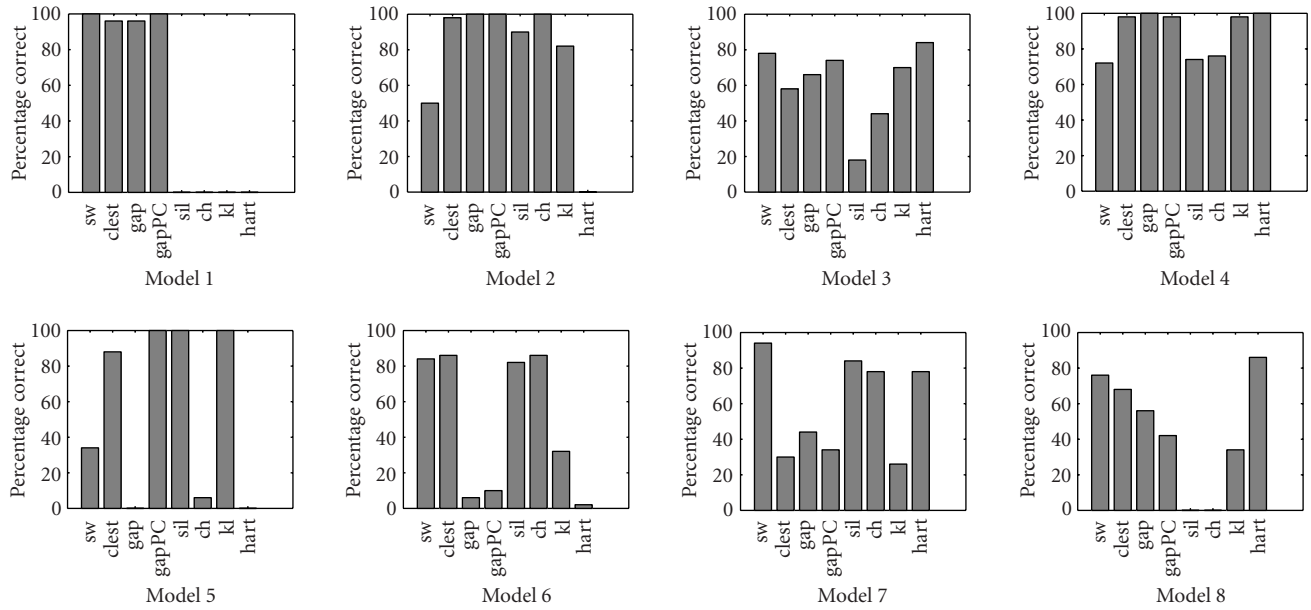


Figure 2: Models 1–8: bar plots representing the percentage of simulations for which the number of clusters is correctly estimated by each method.

clest occurs in Model 7. We emphasize the excellent results of *sw* in finding the two overlapping clusters of Model 7; *clest*, *gap*, and *gapPC* are not able to distinguish between one and two clusters and *kl* overestimates the number of clusters. The behavior of *sw* in Model 2 is surprisingly bad; it is peculiar for Model 2 that the true number of clusters, three, exceeds the dimension of variable space, two. Our interest is on clustering samples from microarray data when N samples (objects) are observed, and each object is associated with a vector of p attributes. Generally, p exceeds by far N, so the number of clusters K is much smaller than the variable space dimension p.

To have a complete picture on the performances of *sw* algorithm, we list in a decreasing order the percentage of simulations for which the number of clusters is correctly estimated by *sw* for every considered model: 100% (Model 1), 94% (Model 7), 84% (Model 6), 78% (Model 3), 76% (Model 8), 72% (Model 4), 50% (Model 2), and 34% (Model 5). The algorithm identifies successfully the lack of structure for Model 1, and for other five structured models, the percentage of correct estimation is larger than 70% which recommends the use of *sw* for a wide family of input data distributions, even if some variables are noisy.

5.2. Clustering the Leukemia dataset

The Leukemia dataset consists of 6817 human genes measured for 72 patients: 47 cases of ALL (38 B-cell ALL and 9 T-cell ALL) and 25 cases of AML. After applying the preprocessing steps described in [3], the measurements for some genes are discarded, and the data are summarized by $N = 72$ vectors in p-dimensional space where $p = 3571$. The results reported in the sequel are obtained without applying any normalization procedure to the data.

We compare the three clusters found by complete-linkage and use all the 3571 genes with the true clusters by displaying in Table 2 the contingency table. We gain more insights by computing the optimal assignment, $A = 26 + 0 + 15 = 41$, according to the definition introduced in Section 3.1. Theorem 1 claims that for inducing identical partitions, we have to remove 31 objects from the dataset, namely, all those objects not associated with any selected entries of the optimal assignment. Since the entry associated to the optimal assignment in the second row has the value zero, the identical induced partitions contain two clusters. This shows that complete-linkage HC amalgamates in C_1 almost all ALL T-cell samples with many ALL B-cell samples, and some AML samples, while in C_2 ALL B-cell samples are grouped with AML samples.

This inability to correctly group the data leads to the conclusion that clustering based on measurements from all genes produces modest results. Therefore we resort to a simple *unsupervised* feature selection method which was also used in [3]: only 100 genes (out of 3571) having the largest variance across tumor samples are employed for clustering. We restrict our investigations to three HC algorithms (group-average, complete-linkage, Ward's method), respectively, three similarity indices ($s(\cdot,\cdot)$, Jaccard, Fowlkes-Mallows), and apply the proposed algorithm when the newly defined space dimension is $p = 100$.

From the dataset consisting of $N = 72$ vectors with length $p = 100$, we select randomly two subsets such that each of them contains 80% of the samples. Then we run the chosen HC algorithm for both subsets and measure the clustering agreement for the samples belonging to the intersection of the subsets. For every hypothesized number of clusters $k \in \{2, 3, \ldots, k_{\max}\}$, the clustering agreement is measured by

Table 7: The estimated number of clusters for Leukemia dataset when measurements from $p = 100$ genes having the largest variance across tumor samples are used. The hypothesized number of clusters varies between 1 and 10. We represent in bold the maximum value over a row.

Hierarchical clustering method	Similarity index	Number of clusters									
		1	2	3*	4	5	6	7	8	9	10
Group-average	$s(\cdot,\cdot)$	0	3	56	40	0	0	0	**184**	17	0
	Jaccard	0	0	0	0	0	0	0	**290**	10	0
	Fowlkes-Mallows	0	0	0	0	0	0	0	**282**	18	0
Complete-linkage	$s(\cdot,\cdot)$	0	75	**212**	12	1	0	0	0	0	0
	Jaccard	0	36	56	**188**	20	0	0	0	0	0
	Fowlkes-Mallows	0	17	31	**207**	37	8	0	0	0	0
Ward's method	$s(\cdot,\cdot)$	0	0	**176**	118	6	0	0	0	0	0
	Jaccard	0	0	95	**202**	3	0	0	0	0	0
	Fowlkes-Mallows	0	0	59	**233**	8	0	0	0	0	0

calculating the value of the selected similarity index *sm*. Repeating the procedure $N_t = 30000$ times, we obtain for every similarity index *sm* and for every allowed value of k, a large set $\mathcal{S}_k \triangleq \{sm_{k,1}, sm_{k,2}, \ldots, sm_{k,N_t}\}$.

The histogram drawn for each k from the corresponding set $\mathcal{S}_k$ plays the key role in the automatic estimation method introduced at the beginning of Section 5. To improve the accuracy, we base the estimation on several histograms for every k. This is performed by splitting each set $\mathcal{S}_k$ into $N_b = 300$ non-overlapping blocks and drawing a different histogram for every block. Observe that the length of a block is $N_\ell = 100$. More precisely, we can write $\mathcal{S}_k = \mathcal{B}_{k,1} \bigcup \mathcal{B}_{k,2} \bigcup \cdots \bigcup \mathcal{B}_{k,N_b}$, where $\mathcal{B}_{k,i} = \{sm_{k,(i-1)\times N_\ell+1}, \ldots, sm_{k,i\times N_\ell}\}$ for $1 \le i \le N_b$. Applying the newly introduced method, we estimate the number of clusters, which is assumed to lie between 1 and $k_{\max}$, using only the blocks $\mathcal{B}_{2,1}, \mathcal{B}_{3,1}, \ldots, \mathcal{B}_{k_{\max},1}$. This is done by computing $m_k = \text{median}(\mathcal{B}_{k,1})$ for $2 \le k \le k_{\max}$ and then applying (7) and (8). Similarly, we obtain another estimation from the blocks $\mathcal{B}_{2,2}, \mathcal{B}_{3,2}, \ldots, \mathcal{B}_{k_{\max},2}$. Continuing the procedure, N_b estimations of the number of clusters are resulting for every pair (*clustering method, similarity index*). For the case $k_{\max} = 10$, we show in Table 7 the distributions of estimated number of clusters when various similarity indices and HC algorithms are applied. For each distribution, we decide that $\hat{K}$ is the value corresponding to the maximum number of occurrences (represented in bold).

According to the existing biological knowledge, the number of clusters for Leukemia dataset is three. Following the procedure described above, we obtain from Table 7 that $\hat{K} = 3$ only when complete-linkage, respectively, Ward's method are used in conjunction with the new similarity index $s(\cdot,\cdot)$. Recall that for the simulated data, only complete-linkage and Ward's method have produced good results. For Leukemia dataset, when these two HC algorithms are applied in combination with Jaccard or Fowlkes-Mallows index, the estimated number of clusters is $\hat{K} = 4$. A possible explanation for $\hat{K} = 4$ relies on the remark, from [7], that ALL B-lineage type samples can be further split into two clusters. Surprisingly, the group-average is leading to $\hat{K} = 8$, which is hard to be given a plausible biological interpretation. The experiments with Leukemia dataset reconfirm that the estimated number of clusters $\hat{K}$ depends strongly on the HC algorithm and on the similarity index. The newly introduced index $s(\cdot,\cdot)$ is the only one that leads to correct estimations.

We further investigate how various HC algorithms cluster the 72×100 Leukemia dataset in classes when Euclidian distance is used to measure the distance between objects. We show in Table 8 the contingency tables when the true partition is compared with partitions produced by clustering algorithms for $\hat{K} \in \{3, 4, 8\}$. In each case, we measure the degree of agreement between the compared partitions by computing the optimal assignment (A^*) as defined in Section 3.1: the larger the value of A^*, the better the degree of agreement. Remark that only the entries of the contingency matrix associated with the optimal assignment (bold represented in Table 8) correspond to samples reliably clustered. The values of A^* reported in Table 8 vary between 45 (group-average) and 53 (complete-linkage), or equivalently, the proportion of reliably clustered samples varies between 63% and 74%.

As expected, A^* declines when the estimated number of clusters $\hat{K}$ is larger than three. For $\hat{K} = 3$, the complete-linkage method clusters properly 30 samples from ALL B-cell class, 8 samples from ALL T-cell class, and 15 samples from AML class. When $\hat{K}$ raises from 3 to 4, only the number of samples from AML class, well classified by complete-linkage method, changes; namely, it decreases from 15 to 12. For $\hat{K} \in \{3, 4\}$, the number of ALL B-cell samples correctly grouped by Ward's method is 28, and 14 AML samples are also well classified. In the case of Ward's method, the number of correctly grouped ALL T-cell samples drops from 8 to 6 when $\hat{K}$ increases from 3 to 4. It is obvious that the smallest A^* is obtained for group-average for which $\hat{K} = 8$; remark in this case that 8 ALL T-cell samples are assigned to the same group.

The importance of feature selection is revealed when comparing the results reported, in Tables 2 and 8, for $\hat{K} = 3$. Using the measurements of only variance-based selected 100 genes improves significantly the clustering. The issue of feature selection for stability-based algorithms is addressed in

Table 8: The contingency tables for the Leukemia dataset: the true partition given by a priori knowledge on the type of disease for each patient is compared with partitions produced by HC algorithms when $\hat{K} \in \{3, 4, 8\}$. Only 100 genes having the largest variance across tumor samples are used for clustering. For each contingency table, the entries associated to the optimal assignment are represented in bold.

Hierarchical clustering method	Cluster	ALL B-cell	ALL T-cell	AML	A^*
Group-average $\hat{K} = 8$	C_1	3	0	**11**	45
	C_2	1	1	3	
	C_3	**26**	0	5	
	C_4	3	0	2	
	C_5	2	0	0	
	C_6	1	0	3	
	C_7	1	0	0	
	C_8	1	**8**	1	
Complete-linkage $\hat{K} = 3$	C_1	**30**	0	8	53
	C_2	3	**8**	2	
	C_3	5	1	**15**	
Complete-linkage $\hat{K} = 4$	C_1	5	1	**12**	50
	C_2	0	0	3	
	C_3	**30**	0	8	
	C_4	3	**8**	2	
Ward's method $\hat{K} = 3$	C_1	**28**	0	7	50
	C_2	5	**8**	4	
	C_3	5	1	**14**	
Ward's method $\hat{K} = 4$	C_1	5	2	4	48
	C_2	0	**6**	0	
	C_3	**28**	0	7	
	C_4	5	1	**14**	

[30], and also a special form, namely, leading principal components selection is investigated in [6]. In this paper, we focus on the choice of the HC algorithm, respectively, the similarity index, and we refer for the feature selection problem to the rich literature on this topic.

6. CONCLUSION

In this study, we present a stability-based method applied for the estimation of the number of clusters in microarray data. To gain insights into the choice of the similarity index and HC algorithm, a careful study on simulated and real data is performed.

A new similarity index $s(\cdot, \cdot)$ is introduced, and its capabilities are evaluated against other well-known similarity indices, based on a benchmark originally proposed in [21]. In this framework, $s(P, P')$ takes small values when partition P' is obtained from partition P after *severe* modifications, which recommends the use of $s(\cdot, \cdot)$ in practical applications. The index $s(\cdot, \cdot)$ is further evaluated in standard experimental conditions when measuring the agreement between the true partition and the partition obtained at each level of an HC solution. We draw the conclusion that a value of 0.8 for $s(\cdot, \cdot)$ is likely to reflect the recovery of some part of the true structure. Moreover, since microarray data are noisy, when necessary to obtain grouping in k clusters, we do not choose automatically the clusters at kth depth in the dendrogram, but move down the hierarchical tree until k nonsingleton clusters are identified.

We note the superiority of $s(\cdot, \cdot)$ and Jaccard when compared to Fowlkes-Mallows index. In experiments with simulated data, the use of $s(\cdot, \cdot)$ was leading to the highest percentage of recovering the true number of clusters five times, while Jaccard index three times and Fowlkes-Mallows index only once. Also for the Leukemia dataset, $s(\cdot, \cdot)$ is the only index which leads to the correct estimation of the number of clusters ($\hat{K} = 3$). We emphasize that the definition of $s(\cdot, \cdot)$ relies on optimal assignment, which is the core of a visualization tool newly proposed in this paper for the interpretation of microarray data clustering.

The good performances of complete-linkage algorithm and Ward's method, observed in Section 5.1 for artificial data, have been reconfirmed for Leukemia data. Even when basing the clustering only on 100 selected genes, the results in Table 8 show the presence of misclassified samples for $\hat{K} = 3$. A major drawback of agglomerative HC was already pointed out in [12]: the fusions once made are irrevocable, so when an algorithm has joined two individuals, they cannot subsequently be separated. A similar drawback occurs for divisive HC algorithms, while partition methods can reconsider, at every stage of clustering, to which group to assign an object [12]. We conclude that agglomerative HC algorithms like

complete-linkage or Ward's method are well suited to be used with the newly introduced method for the estimation of the number of clusters. The resulting method will offer reliable estimates for K and at the same time will be very fast since the HC is computationally efficient; the same tree can be used for all values of $k \in \{2, 3, \ldots, k_{\max}\}$ by looking at different levels of the tree each time. Once K is estimated, partition methods can be further employed for assigning the objects to the clusters.

APPENDICES

A. PROOFS OF PROPOSITIONS

Proof of Proposition 1. It results from the definition that $D(P, P') = D(P', P) \geq 0$ for each pair of partitions (P, P'), while $D(P, P') = 0$ if and only if P and P' are identical. It remains to verify the triangle inequality. Consider three partitions P, P' and P'' of the objects in T. Let U (and V) denote the minimal subset of T which must be removed such that the induced partitions on P and P' (resp., on P' and P'') are identical. Removing $U \bigcup V$ from T induces identical partitions on P, P', and P'', which leads to the chain of inequalities and equalities: $D(P, P'') \leq |U \bigcup V| \leq |U| + |V| = D(P, P') + D(P', P'')$. □

Proof of Proposition 2. Since all entries of M are nonnegative integers and their sum is $N > 0$, there exist at least one entry m_{ij} such that $m_{ij} > 0$. This leads to $A(P, P') \geq 1$ which is equivalent to $D(P, P') \leq N - 1$. When $P = \{P_1, P_2, \ldots, P_N\}$ with $|P_1| = |P_2| = \cdots = |P_N| = 1$ and $P' = \{T\}$, the matrix M reduces to a column vector having only ones as entries, which implies that $D(P, P') = N - 1$. Conversely, when $D(P, P') = N - 1$ and $|P| = r \geq c = |P'|$, let $m_{ij} = 1$ be the only entry of M which is considered in the computation of the optimal assignment $A(P, P') = N - D(P, P') = 1$. Since no entry of the columns with indexes different of j is considered in $A(P, P')$, it follows that all the columns contain only zeros, so, M is essentially a column vector. Because this column vector does not have any entry larger than one, the partition P' consists of a single cluster and the partition P consists only of clusters containing single-objects. □

B. A LOWER BOUND FOR $E[s(\cdot, \cdot)]$ UNDER THE HYPOTHESIS OF GENERALIZED HYPERGEOMETRIC DISTRIBUTION

Proposition B.1. *Under the assumption of fixed margins $m_{i\cdot}$ and $m_{\cdot j}$, and random allocation of matching counts to m_{ij},*

$$E\left[s(P, P')\right] \geq \frac{1}{N-1}\left(\frac{\sum_{i=1}^{c} \alpha_{(i)}\beta_{(i)}}{N} - 1\right) \geq \frac{1}{N-1}\left(\frac{\sum_{i=1}^{c} \alpha_{(i)}}{c} - 1\right), \tag{B.1}$$

where $\alpha_{(1)}, \alpha_{(2)}, \ldots, \alpha_{(r)}$ and $\beta_{(1)}, \beta_{(2)}, \ldots, \beta_{(c)}$ are the elements of the set $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$, respectively, the set $\{\beta_1, \beta_2, \ldots, \beta_c\}$ decreasingly ordered.

Proof. Consider the particular assignment value $a(P, P') \triangleq \sum_{i=1}^{c} m_{(i),(i)}$. By definition, $A(P, P') \geq a(P, P')$, and consequently, $E[A(P, P')] \geq E[a(P, P')]$. This observation, together with definition (3) and $E[a(P, P')] = \sum_{i=1}^{c} \alpha_{(i)}\beta_{(i)}/N$, proves the first inequality in (B.1). The second inequality results from the Chebyshev inequality [31] applied for the sequences $(\alpha_{(1)}, \alpha_{(2)}, \ldots, \alpha_{(c)})$ and $(\beta_{(1)}, \beta_{(2)}, \ldots, \beta_{(c)})$; we also used the identity $\sum_{i=1}^{c} \beta_{(i)} = N$. Note that the equality occurs if and only if $\alpha_{(1)} = \alpha_{(2)} = \cdots = \alpha_{(c)}$ or $\beta_{(1)} = \beta_{(2)} = \cdots = \beta_{(c)}$. □

Corollary B.1. (a) *When $r > c$, the maximum value of the lower bound,*

$$\max_{\alpha_1, \alpha_2, \ldots, \alpha_r} \frac{1}{N-1}\left(\frac{\sum_{i=1}^{c} \alpha_{(i)}}{c} - 1\right) = \frac{1}{c}\frac{N-r}{N-1}, \tag{B.2}$$

is achieved whenever $\alpha_{(c+1)} = \alpha_{(c+2)} = \cdots = \alpha_{(r)} = 1$.

(b) *When $r = c$, the expression of the lower bound becomes $(1/(N-1))(N/c - 1)$.*

C. ASYMPTOTIC AND FINITE SAMPLE CHARACTERISTICS FOR THE SIMILARITY INDICES

We illustrate the computation of $s(P, P')$ by considering an example from [21]: two partitions of six objects, $P = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$ and $P' = \{\{x_1, x_2\}, \{x_3, x_4, x_5\}, \{x_6\}\}$. Elementary calculations lead to $s(P, P') = 0.6$ which is equal to the Rand index value reported in [21]. The same example was used in [25] to compare Rand_{HA}, which takes the value $2/17 \approx 0.1176$, with $\text{Rand}_{MA} = 1/3 \approx 0.3333$. For this particular case, $s(\cdot, \cdot)$ and Rand index take the same value which is larger than the adjusted forms of Rand. We consider in this section more comparisons of the newly introduced index with Rand, Rand_{HA}, Rand_{MA}, Jaccard, and Fowlkes-Mallows indices.

To study the finite and asymptotic characteristics, assume that the original data partition P consists of K clusters with n objects each; ten cases when P' is obtained from P after various simple and major modifications are considered. This approach was firstly proposed in [21] to establish some formal properties of Rand index and further used in [9] when evaluating the performances of the Fowlkes-Mallows index. The expressions of Rand, Rand_{MA}, Jaccard, and Fowlkes-Mallows indices for all the ten cases are given in [23]. We compute in Table C.1 the close forms for the partition distance and the index $s(P, P')$ when P' is obtained by modifying P as described in [21].

We compute also the asymptotics when the number of objects in each cluster increases without bound ($n \to \infty$), while the number of clusters is fixed (K fixed). We observe from the fourth column in Table C.1 that the index asymptotics for the fourth and fifth scenarios are equal to 1.0, which is also true for all similarity indices analyzed in [23]. As it was already pointed out in [23], this is reasonable since P and P' are different in, at most, K points; differences of this magnitude are not very serious if an infinite number of the other points are clustered identically by P and P'.

Table C.1: Expressions for the partition distance $D(\cdot,\cdot)$ and the index $s(\cdot,\cdot)$ between two similar partitions, given an initial partition P which has K clusters of n objects each.

P' is a simple modification of the original partition P				
Modification of P	$D(P,P')$	$s(P,P')$	$\lim_{n\to\infty} s(P,P')$ K fixed	$\lim_{n\to\infty} s(P,P')$ $K=\lambda n$
Two clusters joined	n	$\frac{n(K-1)-1}{nK-1}$	$\frac{K-1}{K}$	1.0
One cluster splits into two equal parts (n even)	$n/2$	$\frac{n(K-1/2)-1}{nK-1}$	$\frac{K-1/2}{K}$	1.0
One cluster splits into single-object clusters	$n-1$	$\frac{n(K-1)}{nK-1}$	$\frac{K-1}{K}$	1.0
One object taken from each cluster to form a new cluster of K objects	K	$\frac{(n-1)K-1}{nK-1}$	1.0	1.0
P' and P'' are similar modifications of the original partition P				
Differences between P' and P''	$D(P',P'')$	$s(P',P'')$	$\lim_{n\to\infty} s(P',P'')$ K fixed	$\lim_{n\to\infty} s(P',P'')$ $K=\lambda n$
Movement of an object to different clusters	1	$\frac{nK-2}{nK-1}$	1.0	1.0
Different clusters split into two equal parts (n even)	n	$\frac{n(K-1)-1}{nK-1}$	$\frac{K-1}{K}$	1.0
Different pairs of clusters joined	$2n$	$\frac{n(K-2)-1}{nK-1}$	$\frac{K-2}{K}$	1.0
P' is a major modification of the original partition P				
Modification of P	$D(P,P')$	$s(P,P')$	$\lim_{n\to\infty} s(P,P')$ K fixed	$\lim_{n\to\infty} s(P,P')$ $K=\lambda n$
All clusters joined into one large cluster	$n(K-1)$	$\frac{n-1}{nK-1}$	$1/K$	0.0
All clusters split into single-object clusters	$(n-1)K$	$\frac{K-1}{nK-1}$	0.0	0.0
n clusters are formed with K objects in each, one object from each original cluster	$nK-\min(n,K)$	$\frac{\min(n,K)-1}{nK-1}$	0.0	0.0

The asymptotic values for $s(P,P')$ and the Jaccard index coincide for seven out of ten evaluated situations, while the asymptotics of Jaccard index never exceed the asymptotics of Fowlkes-Mallows index [23]. Comparing the expressions of Jaccard and Fowlkes-Mallows indices given in Section 3.1, it is easy to prove that the Jaccard index cannot be larger than the Fowlkes-Mallows index when both are well defined. We pay particular attention to the behavior of the similarity indices for the last three scenarios (severe cases). The asymptotics for $s(P,P')$ are 0.0 in the last two cases (identical with the values of Rand_{MA} and Jaccard), which shows the superiority of $s(\cdot,\cdot)$ when comparing with the Rand index. The value reported for Rand in [21] in both cases is $(K-1)/K$, and seems unacceptable since it is too close to 1.0. For the nineth modification, the Fowlkes-Mallows index is not defined, while for the tenth modification, it is equal to 0.0. The asymptotic value of $s(P,P')$ is $1/K$ when the modification is such that all clusters are joined into one large cluster, and being smaller than 0.5 for any $K \geq 2$, it may be considered acceptable. For that case, Rand and Jaccard are also equal to $1/K$, while Fowlkes-Mallows is larger $(1/\sqrt{K})$ and $\text{Rand}_{MA} = 0.0$.

When K is allowed to increase without bound ($K \to \infty$), n must also increase without limit, and the solution is to let K increase as a simple proportion of n ($K = \lambda n$) [9]. The results are reported in the last column of Table C.1: only in the severe cases, the computed value is 0.0, while for other situations is 1.0. The behavior is identical for Rand_{MA}, Jaccard, and Fowlkes-Mallows indices, while Rand is equal to 1.0 for the last two severe cases.

Considering an example based on fixed values for n and K, Table C.2 compares different indices when $n = K = 4$. The severity of the modification from the true clustering is ranked as in [23], where Rand, Rand_{MA}, Fowlkes-Mallows and Jaccard similarity measures have been compared for $n = K = 4$. Rand takes values close to one in many cases, while

Table C.2: Six criteria measures computed for two similar partitions, given an initial partition which has $K = 4$ clusters with $n = 4$ objects each (the largest and second largest values are framed).

Modification of true clustering	Rand	Rand_{HA}	Rand_{MA}	Fowlkes Mallows	Jaccard	$s(\cdot,\cdot)$
Two clusters joined (Serious)	**0.8667**	0.6667	0.7143	0.7746	0.6000	0.7333
One cluster splits into two equal parts (Not Serious)	**0.9667**	0.8889	0.9130	0.9129	0.8333	0.8667
One cluster splits into single-object clusters (Slight Severity)	**0.9500**	0.8276	0.8667	0.8660	0.7500	0.8000
One object taken from each cluster to form a new cluster of k objects (Serious)	**0.8500**	0.4828	0.6000	0.5774	0.4000	0.7333
Movement of an object to different clusters (Not Serious)	**0.9333**	0.7979	0.8367	0.8400	0.7241	**0.9333**
Different clusters split into two equal parts (Slight)	**0.9333**	0.7600	0.8171	0.8000	0.6667	0.7333
Different pairs of clusters joined (Serious)	**0.7333**	0.4000	0.4667	0.6000	0.4286	0.4667
All clusters joined into one large cluster (Severe)	0.2000	0.0000	0.0000	**0.4472**	0.2000	0.2000
All clusters split into single-object clusters (Severe)	**0.8000**	0.0000	0.3333	undefined	0.0000	0.2000
n clusters are formed with k objects in each, one object from each original cluster (Severe)	**0.6000**	−0.2500	0.0000	0.0000	0.0000	0.2000

the indices "corrected for chance" (Rand_{HA} and Rand_{MA}) have always smaller values. We observe that in all cases, Rand_{HA} is smaller than Rand_{MA}. The value of Rand_{HA} in the last row of the table is negative. In general, Rand_{HA} takes values between -1 and 1, but negative values of the index have no substantive use [25]. When the compared partitions are chosen as described in the last row of Table C.2, for any $n = K \geq 2$, the contingency table is an $n \times n$ matrix with all entries equal to one. Simple calculations show that $\text{Rand}_{HA} = -1/n < 0$, which leads to $\text{Rand}_{HA} = -0.25$ for $n = 4$. When n (and implicitly K) is allowed to increase without bound, Rand_{HA} has the limit 0.0.

When the similarity index takes small values for severe cases, then it is recommended to be used in practical applications [23]. Among the considered indices, $s(\cdot,\cdot)$ is the largest only for a modification ranked *not serious*, and the second largest for a *serious* modification and two *severe* modifications. For the case $n = K = 4$, the only index which shows a better behavior is Jaccard.

ACKNOWLEDGMENT

This work has been supported by Academy of Finland Project no. 44876, 80743, and 80745.

REFERENCES

[1] S. P. Smith and R. Dubes, "Stability of a hierarchical clustering," *Pattern Recognition*, vol. 12, pp. 177–187, 1980.

[2] J. N. Breckenridge, "Replicating cluster analysis: method, consistency, and validity," *Multivariate Behav. Res.*, vol. 24, no. 2, pp. 147–161, 1989.

[3] S. Dudoit and J. Fridlyand, "A prediction-based resampling method for estimating the number of clusters in a dataset," *Genome Biology*, vol. 3, no. 7, pp. research 0036.1–0036.21, 2002, http://genomebiology.com/2002/3/7/research/0036.

[4] R. Tibshirani, G. Walther, D. Botstein, and P. Brown, "Cluster validation by prediction strength," Tech. Rep., Department of Biostatistics, Stanford University, September 2001.

[5] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Biocomputing 2002: Proc. Pacific Symposium*, R. B. Altman and K. Lauderdalc, Eds., vol. 7, pp. 6–17, Kauai, Hawaii, USA, 2002.

[6] A. Ben-Hur and I. Guyon, "Detecting stable clusters using principal component analysis," in *Methods in Molecular Biology*, M. J. Brownstein and A. Kohodursky, Eds., pp. 159–182, Humana Press, Totowa, NJ, USA, 2003.

[7] S. Monti, P. Tamayo, J. Mesirov, and T. R. Golub, "Consensus clustering: a resampling-based method for class discovery and vizualization of gene expression microarray data," *J. Mach. Learn. Res.*, vol. 52, no. 1-2, pp. 91–118, 2003.

[8] A. Almudevar and C. Field, "Estimation of single-generation sibling relationships based on DNA markers," *J. Agric. Biol. Environ. Stat.*, vol. 4, no. 1, pp. 136–165, 1999.

[9] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *J. Amer. Statist. Assoc.*, vol. 78, no. 383, pp. 553–569, 1983.

[10] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley, NY, USA, 1990.

[11] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, 1998.

[12] B. S. Everitt, *Cluster Analysis*, Edward Arnold, London, UK, 3rd edition, 1993.

[13] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene

expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[14] F. Azuaje and N. Bolshakova, "Clustering genomic expression data: design and evaluation principles," in *A Practical Approach to Microarray Data Analysis*, D. P. Berrar, W. Dubitzky, and M. Granzow, Eds., pp. 230–245, Kluwer Academic Publishers, Mass, USA, 2002.
[15] N. Bolshakova and F. Azuaje, "Cluster validation techniques for genome expression data," *Signal Processing*, vol. 83, no. 4, pp. 825–833, 2003.
[16] M. Granzow, D. Berrar, W. Dubitzky, A. Schuster, F. Azuaje, and R. Elis, "Tumor classification by gene expression profiling: comparison and validation of five clustering methods," *ACM-SIGBIO Newsletters*, vol. 21, no. 1, pp. 16–22, 2001.
[17] W. Krzanowski and Y. Lai, "A criterion for determining the number of groups in a dataset using sum of squares clustering," *Biometrics*, vol. 44, pp. 23–34, 1985.
[18] J. A. Hartigan, "Statistical theory in clustering," *J. Classification*, vol. 2, pp. 63–76, 1985.
[19] R. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statistics*, vol. 3, no. 1, pp. 1–27, 1974.
[20] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 63 (pt 2), pp. 411–423, 2001.
[21] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.
[22] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Bul. Soc. Vaud. Sci. Nat.*, vol. 44, pp. 223–270, 1908.
[23] G. W. Milligan and D. A. Schilling, "Asymptotic and finite sample characteristics of four external criterion measures," *Multivariate Behav. Res.*, vol. 20, pp. 97–109, 1985.
[24] D. Gusfield, "Partition-distance: a problem and class of perfect graphs arising in clustering," *Inform. Process. Lett.*, vol. 82, no. 3, pp. 159–164, 2002.
[25] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, pp. 193–218, 1985.
[26] L. Morey and A. Agresti, "The measurement of classification agreement: an adjustment to the Rand statistic for the chance agreement," *Educ. Psychol. Meas.*, vol. 44, pp. 33–37, 1984.
[27] G. W. Milligan and M. C. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behav. Res.*, vol. 21, pp. 441–458, 1986.
[28] G. W. Milligan, "An algorithm for generating artificial test clusters," *Psychometrika*, vol. 50, no. 1, pp. 123–127, 1985.
[29] C. Edelbrock, "Mixture model tests of hierarchical clustering algorithms: the problem of classifying everybody," *Multivariate Behav. Res.*, vol. 14, pp. 367–384, 1979.
[30] M. Smolkin and D. Ghosh, "Cluster stability scores for microarray data in cancer studies," *BMC Bioinformatics*, vol. 4, no. 1, pp. 36, 2003.
[31] D. S. Mitrinovic and P. M. Vasic, *Analytic Inequalities*, Springer-Verlag, NY, USA, 1970.

Ciprian Doru Giurcăneanu was born in Birlad, Romania, in 1968. He received the M.S. degree in control engineering from the Department of Control and Computers, "Politehnica" University of Bucharest, Romania, in 1993, and the Ph.D. degree in signal processing (with honors) from the Department of Information Technology, Tampere University of Technology, Finland, in 2001. From 1993 to 1997, he was a Junior Assistant at "Politehnica" University of Bucharest. Since 1997, he has been with the Institute of Signal Processing, Tampere University of Technology, where he is currently a Senior Researcher. From September 2002 to August 2003, he was a Research Fellow with the Academy of Finland. His current research interests include genomics and lossless signal compression.

Ioan Tăbuş received the M.S. degree in control systems and computers in 1982 from "Politehnica" University of Bucharest, Romania, the Ph.D. degree in control systems in 1993 from "Politehnica" University of Bucharest, and the Ph.D. degree in signal processing (with honors) in 1995 from Tampere University of Technology (TUT), Finland. He was a Teaching Assistant from 1984 to 1990, Lecturer from 1990 to 1993, and Associate Professor from 1994 to 1995 in the Department of Control and Computers, "Politehnica" University of Bucharest. From 1996 to 1999, he was a Senior Researcher at TUT. Since January 2000, he has been a Professor in the Institute of Signal Processing at TUT. His research interests include genomic signal processing, speech, audio, image, and data compression, joint source and channel coding, nonlinear signal processing, and image processing. He is the coauthor of two books and more than 90 publications in the fields of signal compression, image processing, and system identification. He is a Senior Member of IEEE and Associate Editor for IEEE Transactions on Signal Processing. He was Chair of IEEE SP/CAS Chapter of Finland Section. Dr. Tabus is a corecipient of 1991 "Traian Vuia" Award of the Romanian Academy and corecipient of the NSIP 2001 Best Paper Award.

EURASIP Journal on Applied Signal Processing 2004:1, 81–91

Segmentation of DNA into Coding and Noncoding Regions Based on Recursive Entropic Segmentation and Stop-Codon Statistics

Daniel Nicorici

Tampere International Center for Signal Processing, Tampere University of Technology, P.O. Box 553, Tampere FIN-33101, Finland
Email: daniel.nicorici@tut.fi

Jaakko Astola

Tampere International Center for Signal Processing, Tampere University of Technology, P.O. Box 553, Tampere FIN-33101, Finland
Email: jaakko.astola@tut.fi

Received 28 February 2003; Revised 15 September 2003

Heterogeneous DNA sequences can be partitioned into homogeneous domains that are comprised of the four nucleotides A, C, G, and T and the stop codons. Recursively, we apply a new entropic segmentation method on DNA sequences using Jensen-Shannon and Jensen-Rényi divergences in order to find the borders between coding and noncoding DNA regions. We have chosen 12- and 18-symbol alphabets that capture (i) the differential nucleotide composition in codons and (ii) the differential stop-codon composition along all the three phases in both strands of the DNA. The new segmentation method is based on the Jensen-Rényi divergence measure, nucleotide statistics, and stop-codon statistics in both DNA strands. The recursive segmentation process requires no prior training on known datasets. Consequently, for three entire genomes of bacteria, we find that the use of nucleotide composition, stop-codon composition, and Jensen-Rényi divergence improve the accuracy of finding the borders between coding and noncoding regions in DNA sequences.

Keywords and phrases: recursive segmentation, DNA sequence, information divergence measures, statistics of stop codons, Bayesian information criterion.

1. INTRODUCTION

The computational identification of genes and coding regions in DNA sequences is a major goal and a long-lasting topic for molecular biology, especially for the human genome project [1, 2]. One of the main goals of the human genome project is to provide a complete list of annotated genes that will be used in the biomedical research. Also, methods for reliable identification of genes in anonymous sequences of DNA can speed the process. A number of such methods exist but their predictive performance for finding genes is still not satisfactory [3]. There are two basic problems in gene finding: detection of protein-binding sites of the genes and detection of regions that code for proteins. These problems still are not satisfactorily solved, and the reliable detection of genes and coding regions in DNA sequences is critical for the success of the computational gene discovery from annotated genome sequences [4]. We address in this study the problem of finding the coding regions in DNA sequences that code for proteins.

Almost everything in the organism of living beings is made of proteins. According to the central dogma that forms the backbone of molecular biology, the DNA codes for the production of messenger RNA (mRNA) during the transcription process. The ribosomes "read" this information and use it for protein synthesis during the translation process.

The main genetic material in the prokaryote and the eukaryote cells is represented by the nucleic DNA molecules that have a well-studied structure. There are four kinds of nucleotides that differ by their nitrogenous bases: adenine (A), cytosine (C), thymine (T), and guanine (G). Along two strands of DNA double helix, a pyrimidine in one chain always faces a purine in the other and only the complementary base pairs T-A and G-C exist. A pyrimidine contains bases T and G, and purine contains bases A and C. Also, there is a large redundancy of the protein-coding regions in DNA that is distributed unevenly. There are $4^3 = 64$ codons to specify only 21 outputs, where 20 are amino acids and one output (stop codon) signals the end of the translation process.

One generic feature of DNA sequences is that their statistical properties are not homogeneously distributed along the sequence [5]. There is evidence of long-range correlations in genomic DNA, and it has been attributed to the presence of complex heterogeneities in the DNA sequences [6, 7, 8]. However, the current biological knowledge about coding regions in DNA is still limited to the structure of the codon and functional sites of the genes. The fact that the composition of the nucleotides for positions inside the codon (periodicity of three nucleotides) is different for the coding regions than the noncoding ones provides a strong signal for detection [9, 10].

Many algorithms have been developed for gene recognition based on three-base periodicity [11, 12, 13, 14], codon-usage measure [2], dicodon-usage measure [15], and position-weight matrix [16]. Fickett [17, 18] presents several algorithms for recognizing complete genes and one algorithm for recognizing coding regions. The accuracy of these algorithms for the complete gene recognition is generally high when they are tested on Guigo's dataset [3], but is not so good for the recently completed genomes of different organisms.

Segmentation methods are computational methods used to identify the homogeneous regions based on entropy measures. They are important for DNA-sequence analysis when identifying the borders between coding and noncoding regions [5, 7, 19, 20]. Also, recursive segmentation of DNA sequences has been used for detecting the existence of the isochores, and CpG islands, detecting replication origin and terminus, and complex patterns such as telomeres, and evaluating the genomic complexity [5, 6]. The Jensen-Shannon divergence is one of the most widely used methods for segmenting DNA sequences [5, 6, 7, 19, 20, 21], and is used for recursively separating DNA sequences in homogenous regions with respect to its neighbors. The criterion for continuing the recursive segmentation process can be based on (i) statistical significance [19, 20, 22], or (ii) Bayesian information criterion (BIC) [5, 6, 7, 21].

In this study, we analyze the recursive entropic segmentation for DNA sequences from different bacteria, but this can be easily extended to other DNA sequences of other organisms. All the bacteria's genomes referred to in this study are available on the site of European Bioinformatics Institute (http://www.ebi.ac.uk/genomes/). In [19], Bernaola-Galvan et al. use a 12-symbol alphabet and Jensen-Shannon divergence for finding the borders between coding and noncoding regions in DNA. The 12-symbol alphabet is based on nucleotide statistics inside codons. It is well known that the coding regions contain stop codons within maximum two phases and noncoding regions contain usually stop codons within all three phases [23]. In order to take into account these statistical properties of coding regions, we use the recursive segmentation algorithm proposed by Bernaola-Galvan et al. [19], a new 18-symbol alphabet that takes into account the nonuniform distribution of stop codons within all three phases, Jensen-Rényi divergence, and a new stopping criterion. The stopping criterion based on BIC for recursive segmentation was proposed by Li [5, 7]. Our approach uses only general statistical properties of coding regions. In this way, the prior training on data sets is avoided and furthermore, the search for additional biological information such as splice and promoter regions may also be avoided. It is noted that such additional information could be incorporated in a more concrete implementation of the algorithm [19]. Consequently, for three entire genomes of bacteria, we find that the use of nucleotide and stop-codon composition, and Jensen-Rényi divergence improve the accuracy of finding the borders between coding and noncoding regions in DNA sequences.

2. STOP-CODON STATISTICS

The distribution of stop codons in DNA coding regions is different than in the noncoding regions. Also, it is well known that the stop codons are strong signals in DNA sequences. In coding regions, the stop codons are usually distributed along two phases (reading frames) with the exception of the stop codon that is in a reading frame and signals the end of a gene. This knowledge is employed implicitly by hidden Markov models used in different gene-finding algorithms [4, 24, 25]. Explicitly, for the first time, the stop-codon statistics is used for recognizing coding regions in studies of Wang et al. [23] and Carpena et al. [26].

Different DNA sequences from different organisms are studied in order to show the distribution of stop codons along all three phases in coding and noncoding regions. There are extracted DNA sequences of different lengths—40, 80, 120, and 160 base pairs (bp)—from the following three randomly chosen prokaryote organisms: *Methanococcus jannaschii* (GenBank acc. L77117), *Chlamydia muridarum* (GenBank acc. AE002160), and *Chlamydophila pneumoniae* (GenBank acc. BA000008). The DNA sequences are taken randomly from coding and noncoding regions of the previous bacteria, and they are not overlapping on the same DNA strand.

Table 1 shows the counts of DNA sequences that have stop codons in one, two, and three phases, and no stop codons in neither of the three phases. There is no DNA coding region with stop codons within all three phases, as is shown in Table 1. We take advantage of this by introducing a new alphabet that considers also the stop-codon statistics and Jensen-Rényi divergence.

Also, in Figure 1, it is shown that the counts of stop-codons along all three phases are increasing rapidly with the length of noncoding regions, and in Figure 2, the counts of the stop codons along three phases are decreasing rapidly with the length of coding regions. Similar observations as in Figures 1, 2, and Table 1 have been used before for the introduction of the stop-codon statistics into the gene-finding field [23].

Figures 3 and 4 show the histograms of the lengths of noncoding and coding DNA regions from bacteria *Methanococcus jannaschii*, *Chlamydia muridarum*, and *Chlamydophila pneumoniae*; none of the coding regions of the three chosen bacteria have the length less than 50 bp, but there exist very short noncoding regions.

Table 1: Distribution of stop codons along phases for coding and noncoding DNA regions.

DNA sequence	Sequence length [bp]	Number of sequences	No stop codons [%]	Stop codons in one phase(s) [%]		
				one	two	three
Coding	40	8000	8.21	44.64	47.15	0
Noncoding	40	8000	5.32	31.08	46.36	17.24
Coding	80	4000	1.23	18.15	80.62	0
Noncoding	80	4000	0.45	6.30	37.80	55.35
Coding	120	2000	0.10	6.85	93.05	0
Noncoding	120	2000	0.30	1.85	22.60	75.25
Coding	160	1400	0	3.36	96.64	0
Noncoding	160	1400	0	0.70	13.20	86.20

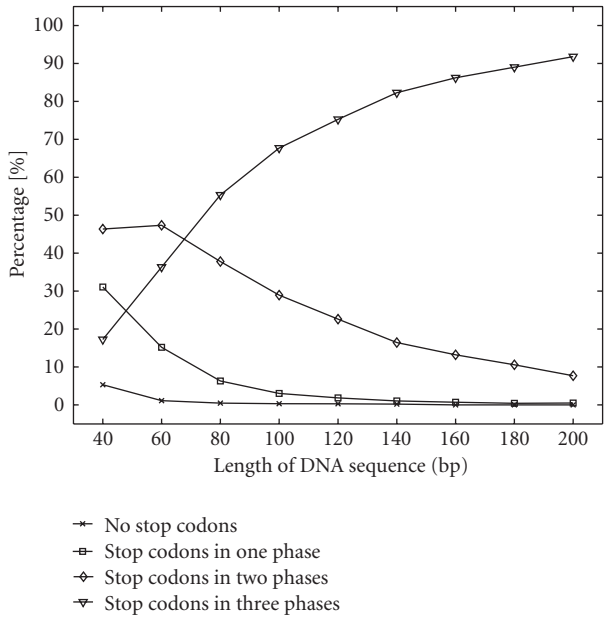


Figure 1: Distribution of stop codons along three phases in noncoding DNA regions.

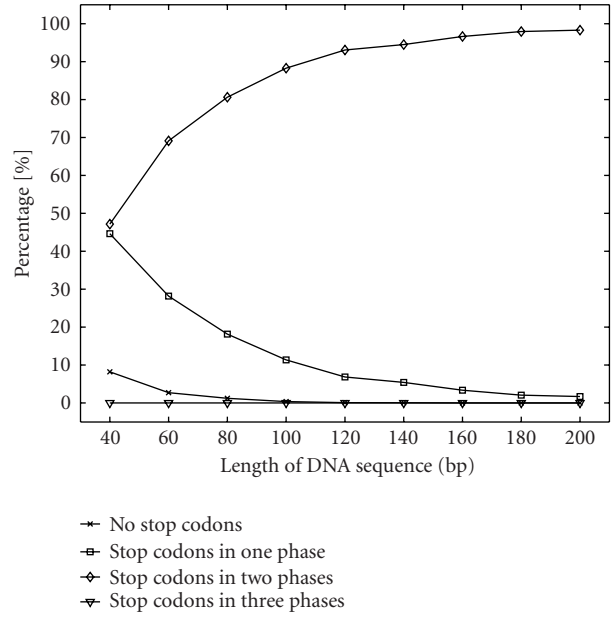


Figure 2: Distribution of stop codons along three phases in coding DNA regions.

The segmentation method based on nucleotide statistics [7] detects the coding regions even when they are on the opposite DNA strand. Thus, the stop-codon statistics along all three phases should also be considered on both DNA strands. As is shown in Figure 5, the stop codons on the reverse DNA strand appear in the given DNA strand, where the stop codons TAA, TAG, and TGA are situated as TCA, CTA, and TTA. When the codon CTA is met on a given DNA strand, it is known that it represents the stop codon TAG on the opposite DNA strand. In this way, the stop-codon statistics in both DNA strands is the same with the statistics of the six codons TAA, TAG, TGA, TCA, CTA, and TAA along a single DNA strand.

3. THE JENSEN-SHANNON DIVERGENCE

The Jensen-Shannon divergence quantifies the difference between two or more probability distributions and is widely used for DNA segmentation [5, 7, 19, 20, 21]. The Jensen-Shannon divergence D_{JS} between m probability distributions $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}$ with the corresponding weights is defined as

$$D_{JS}[\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}] = H\left[\sum_{j=1}^{m} \pi^{(j)} \cdot \mathbf{p}^{(j)}\right] - \sum_{j=1}^{m} \pi^{(j)} \cdot H[\mathbf{p}^{(j)}], \quad (1)$$

where $\mathbf{p}^{(j)} \equiv (p_1^{(j)}, p_2^{(j)}, \ldots, p_k^{(j)})$ are probability distributions satisfying the usual constraints $\sum_{i=1}^{k} p_i^{(j)} = 1$ and $0 \le p_i^{(j)} \le 1$, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, m$; and $\pi^{(j)}$ are the weights of the distributions $\mathbf{p}^{(j)}$, satisfying the constraints $\sum_{j=1}^{m} \pi^{(j)} = 1$ and $0 \le \pi^{(j)} \le 1$. The Shannon entropy of the probability distribution $\mathbf{p}$ used in (1) is defined as

$$H[\mathbf{p}] = -\sum_{i=1}^{k} p_i \cdot \log_2 p_i. \quad (2)$$

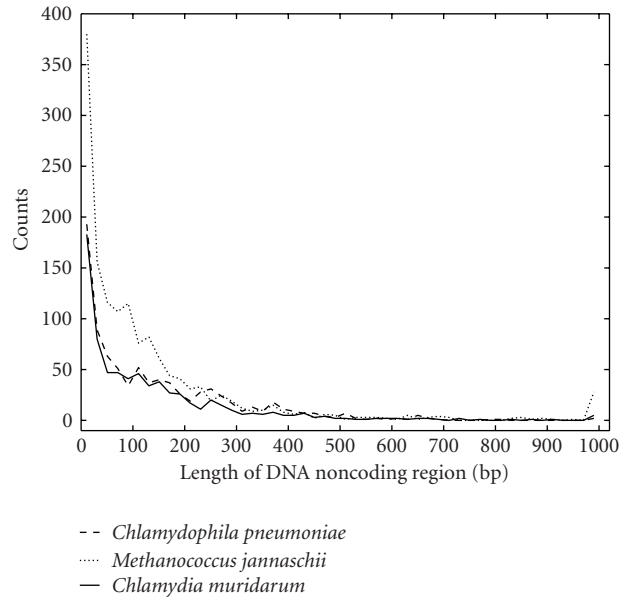


Figure 3: Histograms of the lengths of noncoding DNA regions.

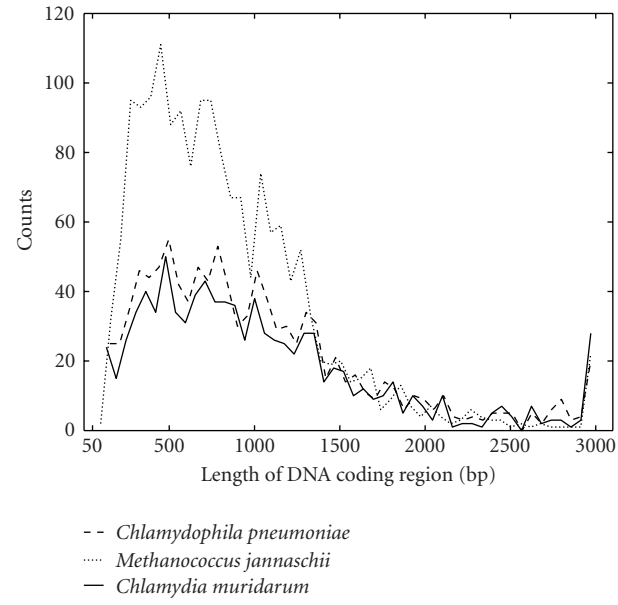


Figure 4: Histograms of the lengths of coding DNA regions.

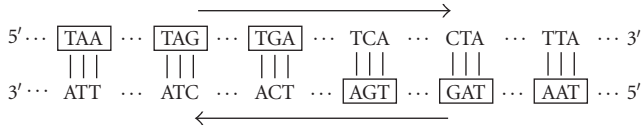


Figure 5: Stop codons in both strands of DNA.

Figure 6 illustrates the three-dimensional representation of the Jensen-Shannon divergence with equal weights for two Bernoulli probability distributions. Some mathematical

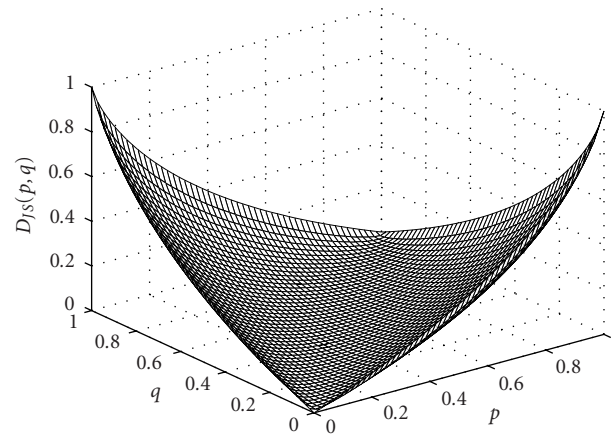


Figure 6: Three-dimensional representation of Jensen-Shannon divergence $D_{JS}(\mathbf{p},\mathbf{q})$, where $\mathbf{p} = (p, 1-p)$, $\mathbf{q} = (q, 1-q)$, and $\pi = (0.5, 0.5)$.

properties for the m-ary case that are important for its application as a divergence measure are the following:

(i) the use of Jensen inequality implies

$$D_{JS}[\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}] \geq 0, \tag{3}$$

where $D_{JS}[\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}] = 0$ if and only if $\mathbf{p}^{(1)} = \mathbf{p}^{(2)} = \cdots = \mathbf{p}^{(m)}$;

(ii) the divergence D_{JS} is symmetric in its arguments $\mathbf{p}^{(1)}$, $\mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}$, that is, is invariant for any permutation of its arguments;

(iii) the divergence D_{JS} is well defined even if $\mathbf{p}^{(1)}$, $\mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}$ are not absolutely continuous.

4. THE JENSEN-RÉNYI DIVERGENCE

The Jensen-Rényi divergence, as Jensen-Shannon divergence, is defined as a similarity measure between two or more probability distributions, and is used in image registration [27]. The Jensen-Rényi divergence D_{JR_α} between m probability distributions $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}$ with the corresponding weights is defined as

$$D_{JR_\alpha}[\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}] = R_\alpha\left[\sum_{j=1}^{m} \pi^{(j)} \cdot \mathbf{p}^{(j)}\right] - \sum_{j=1}^{m} \pi^{(j)} \cdot R_\alpha[\mathbf{p}^{(j)}]. \tag{4}$$

The Rényi entropy of the probability distribution $\mathbf{p}$ referred to in (4) is defined as

$$R_\alpha[\mathbf{p}] = \frac{1}{1-\alpha} \cdot \log_2 \sum_{i=1}^{k} p_i^\alpha, \tag{5}$$

where $\alpha > 0$ and $\alpha \neq 1$. For $\alpha > 1$, the Rényi entropy is neither concave nor convex [27]. For $\alpha \in (0, 1)$, the Rényi

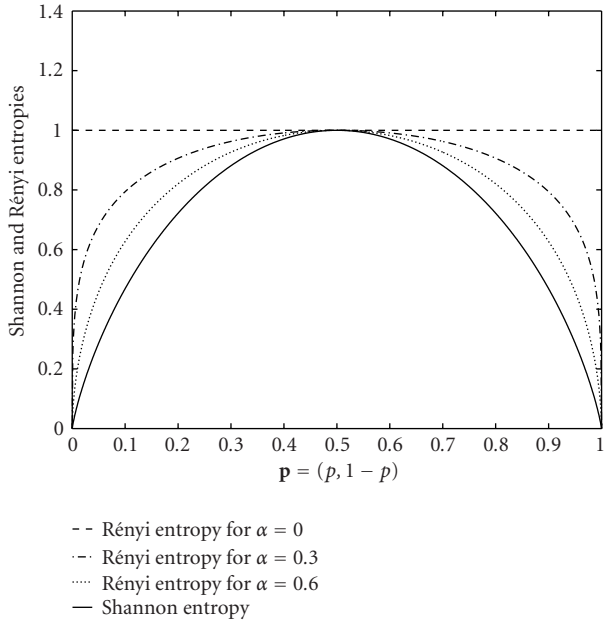


Figure 7: Shannon and Rényi entropies of Bernoulli distribution $\mathbf{p} = (p, 1-p)$ for different values of α.

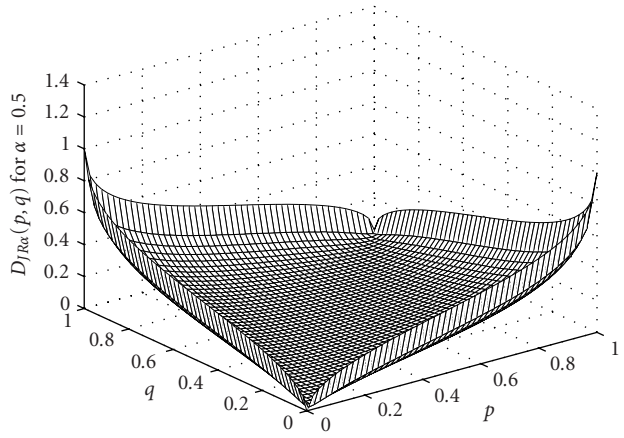


Figure 8: Three-dimensional representation of Jensen-Rényi divergence $D_{JR_\alpha}(\mathbf{p}, \mathbf{q})$, where $\mathbf{p} = (p, 1-p)$, $\mathbf{q} = (q, 1-q)$, $\pi = (0.5, 0.5)$, and $\alpha = 0.5$.

entropy is concave and tends to Shannon entropy $H[\mathbf{p}]$ as $\alpha \to 1$ [27]. The Rényi entropy is a nonincreasing function of α, and thus $R_\alpha[\mathbf{p}] \geq H[\mathbf{p}]$, for all $\alpha \in (0, 1)$. We restrict in this study $\alpha \in (0, 1)$, unless otherwise is specified. As shown in Figure 7, the measure of uncertainty is at a minimum when Shannon entropy is used and it increases as α decreases. The Rényi entropy attains a maximum uncertainty when α is equal to zero [27].

Figure 8 illustrates the three-dimensional representation of the Jensen-Rényi divergence for two Bernoulli probability distributions. Some mathematical properties for the m-ary case, for all $\alpha \in (0, 1)$, that are important for its application as a divergence measure [27] are the following:

(i) the use of Jensen inequality implies

$$D_{JR_\alpha}[\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}] \geq 0, \tag{6}$$

where $D_{JR_\alpha}[\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}] = 0$ if and only if $\mathbf{p}^{(1)} = \mathbf{p}^{(2)} = \cdots = \mathbf{p}^{(m)}$;

(ii) the divergence D_{JR_α} is symmetric in its arguments $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}$, that is, is invariant for any permutation of its arguments;

(iii) the divergence D_{JR_α} is well defined even if $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \ldots, \mathbf{p}^{(m)}$ are not absolutely continuous.

5. DETECTION OF BORDERS BETWEEN CODING AND NONCODING REGIONS USING RECURSIVE SEGMENTATION

We use the approach proposed by Bernaola-Galvan et al. [19, 20] and Li [5, 7] for segmentation of DNA sequences in homogeneous regions that are coding and noncoding. The recursive segmentation of a DNA sequence is as follows. First, the DNA sequence of length N_T is converted into a sequence of symbols with length N using a k-symbol alphabet. We sweep through the symbol sequence, and compute at every position i, where $i = 1, \ldots, N$, that divides the sequence into a left and a right sequence, the entropy of the whole, left, and right sequences. The position where the divergence reaches its maximum is accepted as a cutting point. Further, we recursively apply the segmentation to the left and to the right sequences until the maximized divergence measure is above a certain threshold. For the Jensen-Shannon divergence, the threshold is based on BIC. If the maximized divergence measure is above the threshold, the sequence is segmented, and if not, the segmentation is stopped for the respective sequence.

The Jensen-Shannon divergence D_{JS} is as follows:

$$D_{JS} = \max_i D_{JS}(i) = \left[H - \frac{i}{N}H_l - \frac{N-i}{N}H_r\right], \tag{7}$$

where H, H_l, and H_r are the Shannon entropies (2) of the whole, left, and right sequences, respectively [5, 7, 19, 20]. The weights are i/N and $(N-i)/N$ for the left and right sequences, respectively, where i is the point that divides the sequences into two sequences. In his study, Grosse et al. [22] shows that Jensen-Shannon divergence, as introduced previously, can be interpreted as the mutual information in the framework of information theory.

The Jensen-Rényi divergence D_{JR_α} is as follows:

$$D_{JR_\alpha} = \max_i D_{JR_\alpha}(i) = \left[R_\alpha - \frac{i}{N}R_{\alpha,l} - \frac{N-i}{N}R_{\alpha,r}\right], \tag{8}$$

where R_α, $R_{\alpha,l}$, and $R_{\alpha,r}$ are the Rényi entropies (5) of the whole, left, and right sequences, respectively.

Bernaola-Galvan et al. [19] introduces a 12-symbol alphabet in order to take into account the differential nucleotide composition in codons. The phase of the nucleotide, for this alphabet, is defined as $m = (n \bmod 3) + 1$, where $m \in \{1, 2, 3\}$, and n is the position of the nucleotide in the

Table 2: Symbol mapping for 12-symbol alphabet.

Nucleotide	Phase	Symbol
A	1	A_1
	2	A_2
	3	A_3
C	1	C_1
	2	C_2
	3	C_3
G	1	G_1
	2	G_2
	3	G_3
T	1	T_1
	2	T_2
	3	T_3

Table 3: Stop-codon mapping for 18-symbol alphabet.

Triplets of nucleotides (codons)	Phase	Symbol
TGA, TAG, or TAA	1	S_1
	2	S_2
	3	S_3
TCA, CTA, or TTA	1	S'_1
	2	S'_2
	3	S'_3

DNA sequence. Each nucleotide of the DNA sequence is substituted by the symbols from $\mathcal{A}_{12} = \{A_1, A_2, A_3, C_1, C_2, C_3, G_1, G_2, G_3, T_1, T_2, T_3\}$, as is also shown in Table 2.

We introduce in this study an 18-symbol alphabet that takes into account also the nonuniform distribution of stop-codons in both DNA strands, along all three phases [23]. Thus, the nucleotides and the stop codons are substituted by the symbols from $\mathcal{A}_{18} = \{A_1, A_2, A_3, C_1, C_2, C_3, G_1, G_2, G_3, T_1, T_2, T_3, S_1, S_2, S_3, S'_1, S'_2, S'_3\}$, where the symbols for nucleotides are as for $\mathcal{A}_{12}$ alphabet (Table 2). The symbols S_1, S_2, and S_3 are the stop codons TAA, TAG, and TGA in the given DNA strand, and S'_1, S'_2, and S'_3 are the stop codons AGT, GAT, and AAT on the opposite DNA strand, as shown in Table 3. The phase of a stop codon is defined the same as for a nucleotide with the exception that n represents the position of the first nucleotide of the given codon. For example, the DNA sequence ACTTAA is converted using the 18-symbol alphabet as $A_1C_2S'_3T_3S_1T_1A_2A_3$.

These two alphabets, together with the two divergence measures, are used for finding the borders between coding and noncoding regions in different DNA sequences from bacterium *Rickettsia prowazekii*, as shown in Figures 9 and 10.

In Figure 9, we plot the D_{JR_α} ($\alpha = 0.5$) and D_{JS} with $\mathcal{A}_{12}$ and $\mathcal{A}_{18}$ alphabets along a DNA sequence. The DNA sequence is composed of two randomly chosen regions from bacterium *Rickettsia prowazekii*. The first region of 1016 bp belongs to a coding region and the second one of 1151 bp belongs to a noncoding region. Figure 9 shows that using both divergences and both alphabets, we are able to find the border between the coding and noncoding region. Using $\mathcal{A}_{12}$ alphabet with both divergences, the cut is found at 11 bp to the right of the real border, and using $\mathcal{A}_{18}$ alphabet with both divergences, the cut is found at 4 bp to the left of the real border. In Figure 10, we plot both divergences using the both alphabets along a DNA sequence that contains a coding region of 810 bp from gene **RP172** followed by the original noncoding region of 1477 bp as it appears in the chromosome of bacterium *Rickettsia prowazekii.*

In Table 4, we analyze the same DNA sequence as in Figure 10 and it can be seen that using the alphabet $\mathcal{A}_{18}$ and the Jensen-Rényi divergence, we get the closest cut to the real border between the coding and noncoding regions. When the segmentation is applied on a single continuous DNA sequence followed by the "original" noncoding region as in Figure 10, using the alphabet $\mathcal{A}_{12}$ is not anymore possible to detect with a reasonable accuracy the border between the two regions, because the coding region "leaks," for a small portion, into the noncoding region. The region where the leaking phenomena happens has the same nucleotide composition as a coding region even though it is a noncoding region. This region does not have the same stop-codons composition as a coding region and because of this, using $\mathcal{A}_{18}$ alphabet, we are able to find a much closer border to the real one. The "leaking" regions appear usually in vicinity of the coding regions and they are removed in the cases when two randomly chosen, coding and noncoding, regions are joined arbitrary together, as in Figure 9. The Jensen-Rényi divergence takes better advantage of the $\mathcal{A}_{18}$ alphabet than Jensen-Shannon divergence because the counts of the stop-codons are much less than the counts of the nucleotides. The Jensen-Rényi divergence emphasizes better the difference between the regions with different stop-codon statistics. Thus, using the $\mathcal{A}_{18}$ alphabet and Jensen-Rényi divergence, we are able to detect better the border due to the introduction of the biological knowledge in the segmentation method.

6. STOPPING CRITERION FOR RECURSIVE SEGMENTATION

The stopping criterion in the case of Jensen-Shannon divergence can be considered from the point of view of the hypothesis testing and the model selection framework. For the hypothesis testing framework, the probability that the value of D_{JS} can be obtained by chance is computed by the null hypothesis that the sequence is homogeneous. The exact form of the null distribution is difficult to find [5, 28] but Grosse et al. [9, 22] suggest an empirical form of the null distribution based on numerical simulation.

In this study, the stopping criterion for segmentation using Jensen-Shannon divergence is based on model selection that has been introduced by Li in his studies [5, 7]. The model is judged by how well it fits the data and how complex it is. Thus the stopping criterion tests if a two-random-subsequence model is better than the one-random-sequence

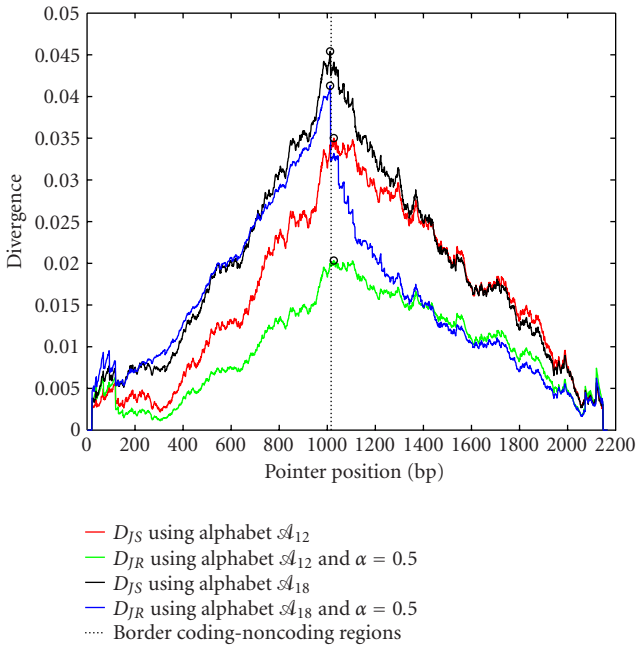


Figure 9: Jensen-Shannon divergence and Jensen-Rényi divergence versus cutting position for a DNA sequence containing a randomly chosen coding region and a randomly chosen noncoding region. The maximum values for the divergences are circled on the graph.

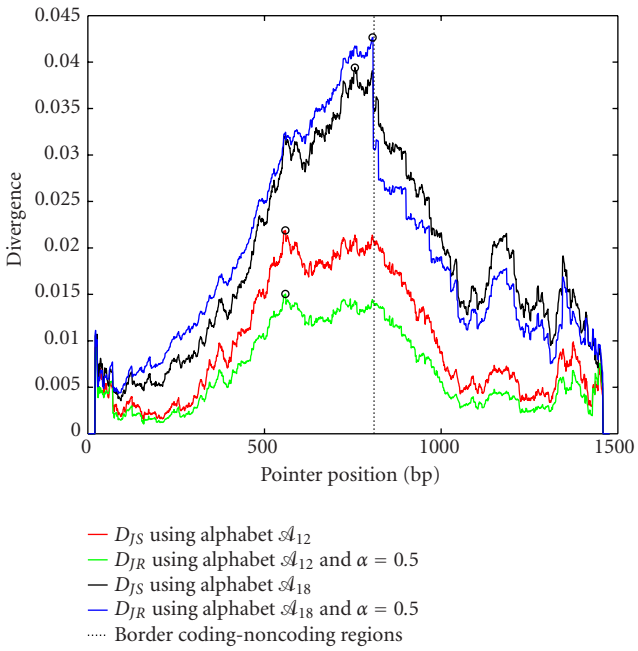


Figure 10: Jensen-Shannon divergence and Jensen-Rényi divergence versus cutting position for a DNA sequence containing a coding region followed by a noncoding region. The maximum values for the divergences are circled on the graph.

model. If the two-random-subsequence model is better, then the cut will be accepted, otherwise it is not. For balancing the goodness-of-fit of the model to the data with the number of parameters, the BIC is used as follows:

$$\Delta\text{BIC} = -2 \cdot \log L + K \cdot \log_2 N, \tag{9}$$

where $L = L_2/L_1$, L_1 and L_2 are the maximum likelihood of the models before and after the cut is made, respectively; $K = K_2 - K_1$, K_1 and K_2 are the number of free parameters before and after the cut is made, respectively; and N is the length of the sequence [5, 7]. In order to continue the recursive segmentation procedure and to decide if a cut is significant or not, the BIC should be reduced, that is, $\Delta\text{BIC} < 0$. This leads to

$$2 \cdot N \cdot D_{JS} > K \cdot \log_2 N. \tag{10}$$

In order to decide when the segmentation algorithm using D_{JS} has to be stopped, Li [5, 7] introduced, as a measure, the segmentation strength as

$$s = \frac{2 \cdot N \cdot D_{JS} - K \cdot \log_2 N}{K \cdot \log_2 N}. \tag{11}$$

The BIC stopping criterion is introduced here only for Jensen-Shannon divergence. In order to decide when the segmentation algorithm using D_{JR_α} has to be stopped, we introduce a new segmentation strength, derived empirically, as

$$s = \frac{2 \cdot N \cdot D_{JR_\alpha} - K \cdot \log_2 N}{K \cdot \log_2 N}. \tag{12}$$

The recursive segmentation continues, or a cut is accepted as significant as long as $s \geq s_0$, where s_0 can be set by the user. By setting the s_0, one affects the threshold used to make the decision if a cut is significant or not. For the $\mathcal{A}_{12}$ and $\mathcal{A}_{18}$ alphabets, the segmentation strength is defined by (11) or (12), where $K = 10$ and $K = 16$, respectively. The segmentation strengths for D_{JS} and D_{JR_α} have a closely related expression. Special cases of Jensen-Rényi divergence are obtained for $\alpha = 1/2$ for which one obtains the log Hellinger distance squared and for $\alpha = 1$ for which one obtains the Kullback-Liebler divergence [29]. For $\alpha = 1$, one obtains $D_{JR_\alpha} = D_{JS}$.

In this study, the standard stopping criterion is the stopping criterion where a cut is accepted as significant as long as $s \geq s_0$, where s is the segmentation strength in (11) and (12). A DNA sequence that does not have stop codons along all three phases has a very high probability (Figures 1 and 2) to be a coding region, and in this case it does not need to be

Table 4: Cuts obtained using different methods for segmentation for the same DNA sequence as in Figure 10.

Segmentation method	Distance from border
D_{JS} with $\mathcal{A}_{12}$ alphabet	251 bp (left)
D_{JR} ($\alpha = 0.5$) with $\mathcal{A}_{12}$ alphabet	251 bp (left)
D_{JS} with $\mathcal{A}_{18}$ alphabet	54 bp (left)
D_{JR} ($\alpha = 0.5$) with $\mathcal{A}_{18}$ alphabet	4 bp (left)

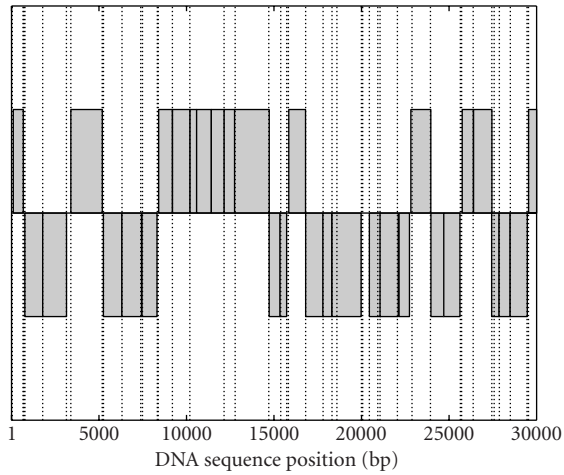


Figure 11: Comparison between the known coding regions (gray regions with solid lines as borders) of a DNA sequence from bacteria *Borrelia burgdorferi* and the borders (vertical dashed lines) obtained through recursive segmentation using Jensen-Rényi divergence ($\alpha = 0.5$), $\mathcal{A}_{18}$ alphabet, and standard stopping criterion. The coding regions oriented downwards are situated on the opposite DNA strand.

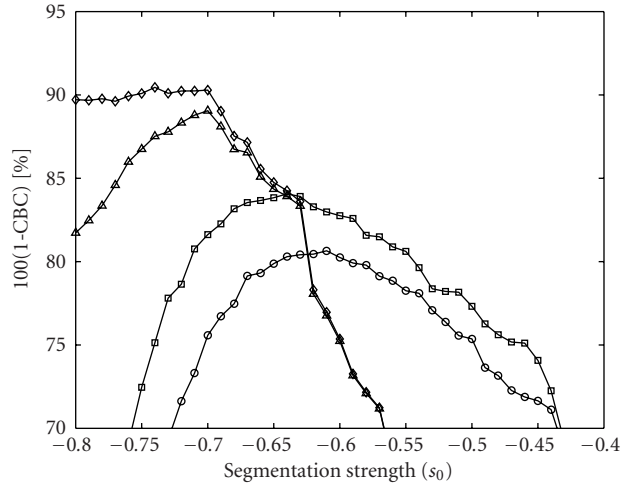


Figure 12: Accuracies of recursive segmentation for different thresholds of segmentation strength using Jensen-Shannon and Jensen-Rényi divergences with $\mathcal{A}_{12}$ and $\mathcal{A}_{18}$ alphabets and two stopping criterions for the genome of bacterium *Rickettsia prowazekii*.

segmented further. Thus, we introduce a new stopping criterion as follows. A cut is accepted as significant if $s \geq s_0$ and the segmented sequence has stop codons in all three phases. Hence, a DNA sequence is not segemented further if it has stop codons only in two phases.

In this study, the DNA sequences smaller than 40 bp in length are not segmented further in the recursive segmentation process because we consider that it is not statistically enough to separate them into two subsequences with a high confidence and the stop-codon statistics is not anymore relevant for such small sequences, as shown in Table 1.

7. EXPERIMENTAL RESULTS

In order to quantify the coincidence between cuts (CBC) obtained using the recursive segmentation algorithm and known borders between coding and noncoding regions, we use the following measure, introduced by Bernaola-Galvan et al. [19]:

$$\mathrm{CBC} = \frac{1}{2}\left[\sum_i \frac{\min_j |b_i - c_j|}{N_T} + \sum_j \frac{\min_i |b_i - c_j|}{N_T}\right], \quad (13)$$

where $\{b_i\}$ is the set of all borders between coding and noncoding regions, $\{c_j\}$ is the set of all cuts produced by the segmentation, and N_T represents the total length of the DNA sequence. The measure CBC is the average of the error in the determination of the correct boundaries between coding and noncoding regions, so the value $(1 - \mathrm{CBC})$ is a reasonable measure of the accuracy of the borders detected between coding and noncoding regions [19].

In Figure 11, a comparison is shown between the known regions of a DNA sequence containing the first 30000 bp from the beginning of the genome of bacterium *Borrelia burgdorferi* and the predicted borders obtained through recursive entropic segmentation using Jensen-Rényi divergence with the $\mathcal{A}_{18}$ alphabet and standard stopping criterion. The threshold of the segmentation strength is $s_0 = -0.55$ where the parameter CBC achieves its overall minimum. The borders between coding and noncoding regions are detected very close to the real ones as shown in Figure 11.

We show in Figures 12, 13, and 14 the results of the recursive segmentation for different values of the segmentation strength—using Jensen-Shannon and Jensen-Rényi divergences with alphabets $\mathcal{A}_{12}$ and $\mathcal{A}_{18}$, and two stopping criterions—of the whole genomes of the bacteria *Rickettsia prowazekii* (GenBank acc. AJ235269, length 1111523 bp), *Borrelia burgdorferi* (GenBank acc. AE000783, length 910724 bp), and *Methanococcus jannaschii* (GenBank acc. L77117, length 1664970 bp). For recursive segmentation of all three genomes with Jensen-Rényi divergence and $\mathcal{A}_{18}$ alphabet, we use $\alpha = 0.5$. This value has been found by segmenting the whole genome of bacterium *Rickettsia prowazekii*, using standard stopping criterion, for $\alpha = 0, 0.1, 0.2, \ldots, 0.9, 1$ and choosing the value for α, where the maximum of segmentation accuracy occurs. The recursive segmentation, using Jensen-Rényi divergence with $\mathcal{A}_{12}$ alphabet, achieves the maximum of the accuracy for $\alpha = 1$ that is the same as Jensen-Shannon divergence. Hence, the Jensen-Rényi divergence takes better advantage of the introduction of the stop-codon statistics than the Jensen-Shannon divergence does.

The recursive segmentation using the Jensen-Rényi divergence with $\mathcal{A}_{18}$ alphabet and new segmentation criterion

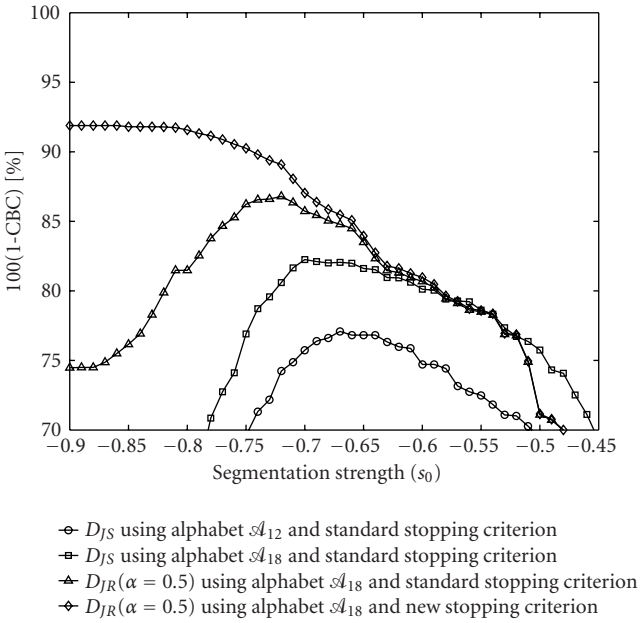


Figure 13: Accuracies of recursive segmentation for different thresholds of segmentation strength using Jensen-Shannon and Jensen-Rényi divergences with $\mathcal{A}_{12}$ and $\mathcal{A}_{18}$ alphabets and two stopping criterions for the genome of bacterium *Borrelia burgdorferi*.

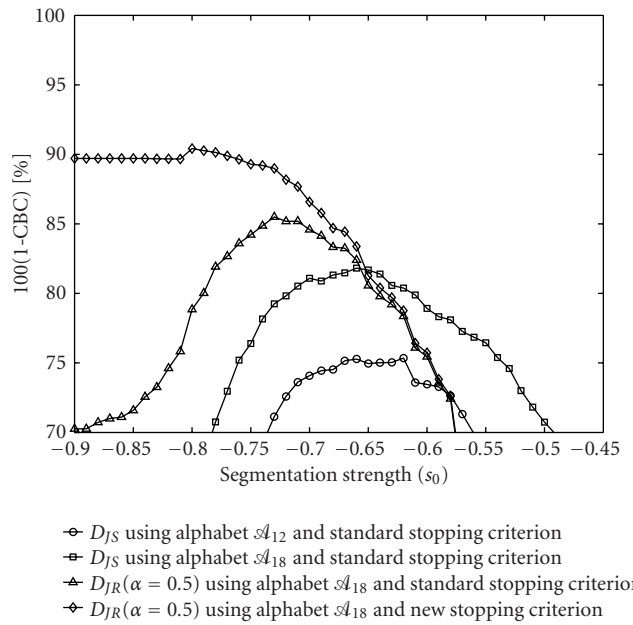


Figure 14: Accuracies of recursive segmentation for different thresholds of segmentation strength using Jensen-Shannon and Jensen-Rényi divergences with $\mathcal{A}_{12}$ and $\mathcal{A}_{18}$ alphabets and two stopping criterions for the genome of bacterium *Methanococcus jannaschii*.

achieves the best overall maximum accuracies for the whole genome of the three bacteria. Bernaola-Galvan et al. [19] achieves the maximum of accuracy in detecting the borders of 80% compared with our 80% with the same Jensen-Shannon divergence and same $\mathcal{A}_{12}$ alphabet. We use the standard stopping criterion based on BIC, compared with the statistical significance used by Bernaola-Galvan et al. [19]. Our newly introduced segmentation method that uses Jensen-Rényi divergence with $\mathcal{A}_{18}$ alphabet and the new stopping criterion gives an accuracy of 90% for $s_0 = -0.74$, that is, higher than 80% reported by Bernaola-Galvan et al. [19]. Also the accuracies for bacteria *Borrelia burgdorferi* and *Methanococcus jannaschiie* are improved from 77% and 75% with Jensen-Shannon divergence using $\mathcal{A}_{12}$ alphabet and standard stopping criterion to 91% and 89% with Jensen-Rényi divergence using $\mathcal{A}_{18}$ alphabet and new stopping criterion, respectively. The improvement in accuracy is explained by the use of Jensen-Rényi divergence that takes better advantage of the stop-codon statistics than Jensen-Shannon divergence does. Also, the introduction of the new stopping criterion in this study improves the accuracies of the segmentation. From Figures 12, 13, and 14, a good value of the threshold for the segmentation strength is $s_0 = -0.75$ for segmenting other genomes of bacteria with Jensen-Rényi divergence ($\alpha = 0.5$) using $\mathcal{A}_{18}$ alphabet and new stopping criterion. Even though, for $s_0 > -0.75$, higher accuracies can be achieved in some situations, this is not always true due to the scattering of coding regions in genome.

Consequently, our results that use the newly introduced approach, based on Jensen-Rényi divergence with the $\mathcal{A}_{18}$ alphabet and new stopping criterion, appear to be more accurate than those obtained using only Jensen-Shannon divergence with $\mathcal{A}_{12}$ alphabet and standard stopping criterion, in finding the borders between coding and noncoding regions.

8. DISCUSSION

In this study, we introduce a new segmentation method based on Jensen-Rényi divergence, an 18-symbol alphabet, and a new stopping criterion for finding the borders between coding and noncoding regions. The new segmentation method applied to three bacteria genome improves the accuracies of the border detection compared to the standard segmentation procedures previously reported. We employ the composition of stop codons over all three phases along the DNA sequence in the 18-symbol alphabet and in the new stopping criterion for improving the accuracy of finding the borders between coding and the noncoding DNA regions.

The assumptions built in other gene-finding systems as GENMARK, VEIL [25], and MORGAN [30] have a number of shortcomings [30] that do not affect the recursive entropic segmentation in finding the borders between coding and noncoding regions. A direct comparison between gene-finding and recursive segmentation for finding the borders between coding and noncoding regions is difficult to make because the gene-finding systems perform very well on small DNA sequence that contains only one gene or very few coding regions. The recursive entropic segmentation performs better on long DNA sequences with a large number of genes, in order to gain statistics. The present segmentation

algorithms [5, 7, 19] rely heavily on statistical properties for finding the coding, noncoding, and other regions of interests in DNA, but the gene-finding systems [4, 25, 30] use biological knowledge regarding functional sites, together with statistics for finding genes. Also, the recursive segmentation needs no prior training compared with gene-finding systems that require extensive training on known datasets. In eukaryotes are much more short coding-regions that are more "scattered" than in prokaryotes and thus it is more difficult to find their borders-based statistical properties as in [5]. The genomes analyzed in this study belong only to prokaryotes that have the coding regions much more compact than in eukaryotes.

9. CONCLUSION

There is an increasing need to develop new algorithms for finding coding regions in DNA sequences. In this study, we introduce a new segmentation method based on Jensen-Rényi divergence with an 18-symbol alphabet and new stopping criterion for finding the borders between coding and noncoding regions in prokaryotes. We use recursive segmentation along with a stopping criterion based on Bayesian information criterion (BIC). Together, they offer a novel method to view the compositional heterogeneity of a DNA sequence. The success comes from the utilization of the stop-codon statistics in all three phases along the DNA sequence and use of Jensen-Rényi divergence. For three entire genomes of bacteria, we found that the use of Jensen-Rényi divergence, nucleotide composition, and stop-codon composition improves the accuracy of finding the borders between coding and noncoding regions in DNA sequences, compared to the standard segmentation procedures previously reported.

REFERENCES

[1] J. W. Fickett, "Recognition of protein coding regions in DNA sequences," *Nucleic Acids Research*, vol. 10, no. 17, pp. 5303–5318, 1982.

[2] R. Staden and A. D. McLachlan, "Codon preference and its use in identifying protein coding regions in long DNA sequences," *Nucleic Acids Research*, vol. 10, pp. 141–156, 1982.

[3] M. Burset and R. Guigo, "Evaluation of gene structure prediction programs," *Genomics*, vol. 34, no. 3, pp. 353–367, 1996.

[4] D. Nicorici, J. Astola, and I. Tabus, "Computational identification of exons in DNA with a hidden Markov model," in *Workshop on Genomic Signal Processing and Statistics*, Raleigh, NC, USA, October 2002.

[5] 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, no. 5, pp. 491–510, 2002.

[6] R. K. Azad, J. S. Rao, W. Li, and R. Ramaswamy, "Simplifying the mosaic description of DNA sequences," *Phys. Rev. E*, vol. 66, no. 031913, pp. 1–6, 2002.

[7] W. Li, "New stopping criteria for segmenting DNA sequences," *Phys. Rev. Lett.*, vol. 86, no. 25, pp. 5815–5818, 2001.

[8] P. D. Cristea, "Large scale features in DNA genomic signals," *Signal Processing*, vol. 83, no. 4, pp. 871–888, 2003.

[9] I. Grosse, H. Herzel, S. V. Buldyrev, and H. E. Stanley, "Species independence of mutual information in coding and noncoding DNA," *Phys. Rev. E*, vol. 61, no. 5, pp. 5624–5629, 2000.

[10] W. Li, G. Stolovitzky, P. Bernaola-Galvan, and J. L. Oliver, "Compositional heterogeneity within, and uniformity between, DNA sequences of yeast chromosomes," *Genome Research*, vol. 8, no. 9, pp. 916–928, 1998.

[11] A. A. Tsonis, J. B. Elsner, and P. A. Tsonis, "Periodicity in DNA coding sequences: Implications in gene evolution," *Journal of Theoretical Biology*, vol. 151, no. 3, pp. 323–331, 1991.

[12] S. Tiwari, S. Ramachandran, S. Bhattacharya, A. Bhattacharya, and R. Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences," *CABIOS*, vol. 13, no. 3, pp. 263–270, 1997.

[13] D. Anastassiou, "DSP in Genomics," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 1053–1056, Salt Lake City, Utah, USA, May 2001.

[14] P. P. Vaidyanathan and B.-J. Yoon, "Gene and exon prediction using allpass-based filters," in *Workshop on Genomic Signal Processing and Statistics*, Raleigh, NC, USA, October 2002.

[15] R. Farber, A. Lapedes, and K. Sirotkin, "Determination of eukaryotic protein coding regions using neural networks and information theory," *J. Mol. Biol.*, vol. 226, pp. 471–479, 1992.

[16] R. Staden, "Computer methods to locate signals in nucleic acid sequences," *Nucleic Acids Research*, vol. 12, no. 1, pp. 505–519, 1984.

[17] J. W. Fickett, "Finding genes by computer: the state of the art," *Trends in Genetics*, vol. 12, no. 8, pp. 316–320, 1996.

[18] J. W. Fickett, "The gene identification problem: an overview for developers," *Computer and Chemistry*, vol. 20, no. 1, pp. 103–118, 1996.

[19] 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," *Phys. Rev. Lett.*, vol. 85, no. 6, pp. 1342–1345, 2000.

[20] P. Bernaola-Galvan, R. Roman-Roldan, and J. L. Oliver, "Compositional segmentation and long-range fractal correlations in DNA sequences," *Phys. Rev. E*, vol. 53, no. 5, pp. 5181–5189, 1996.

[21] D. Nicorici, J. A. Berger, J. Astola, and S. K. Mitra, "Finding borders between coding and noncoding DNA regions using recursive segmentation and statistics of stop codons," in *Proceedings of the 2003 Finnish Signal Processing Symposium*, pp. 231–235, Tampere, Finland, May 2003.

[22] I. Grosse, P. Bernaola-Galvan, P. Carpena, R. Roman-Roldan, J. L. Oliver, and H. E. Stanley, "Analysis of symbolic sequences using the Jensen-Shannon divergence," *Phys. Rev. E*, vol. 65, no. 041905, pp. 1–16, 2002.

[23] Y. Wang, C. T. Zhang, and P. Dong, "Recognizing shorter coding regions of human genes based on the statistics of stop codons," *BioPolymers*, vol. 63, no. 3, pp. 207–216, 2002.

[24] M. Borodovsky and J. McIninch, "GENMARK: parallel gene recognition for both DNA strands," *Computer and Chemistry*, vol. 17, no. 2, pp. 123–134, 1993.

[25] J. Henderson, S. Salzberg, and K. H. Fasman, "Finding genes in DNA with a hidden Markov model," *Journal of Computational Biology*, vol. 4, no. 2, pp. 127–141, 1997.

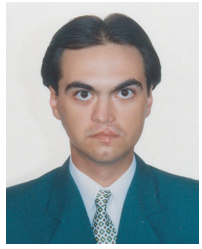
[26] P. Carpena, P. Bernaola-Galvan, R. Roman-Roldan, and J. L. Oliver, "A simple and species-independent coding measure," *Gene*, vol. 300, no. 1–2, pp. 97–104, 2002.

[27] Y. He, A. B. Hamza, and H. Krim, "A generalized divergence measure for robust image registration," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1211–1220, 2003.

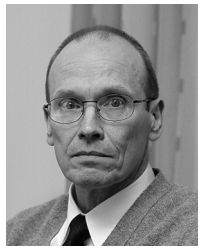
[28] A. N. Pettitt, "A simple cumulative sum type statistic for the change-point problem with zero-one variables," *Biometrika*, vol. 67, no. 1, pp. 79–84, 1980.

[29] A. O. Hero and O. J. J. Michel, "Rényi information divergence via measure transformations on minimal spanning trees," in *Proc. IEEE 2000 International Symposium on Information Theory*, p. 414, Sorrento, Italy, June 2000.

[30] S. Salzberg, A. Delcher, K. Fasman, and J. Henderson, "A decision tree system for finding genes in DNA," *Journal of Computational Biology*, vol. 5, no. 4, pp. 667–680, 1998.



Daniel Nicorici received his B.S. and M.S. degrees in electrical engineering from Technical University of Cluj-Napoca, Romania, in 1999 and 2000, respectively. Since 2001, he has been with Tampere University of Technology, Finland, as a Researcher. He is currently pursuing his Ph.D. at Tampere International Center for Signal Processing. His research interest focuses on genomic signal processing.



Jaakko Astola (IEEE Fellow) received his B.S., M.S., Licentiate, and Ph.D. degrees in mathematics (specialising in error-correcting codes) from Turku University, Finland, in 1972, 1973, 1975, and 1978, respectively. From 1976 to 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, Lappeenranta, 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 to 1992, he was an Associate Professor in applied mathematics at Tampere University, Tampere, Finland. Since 1993, he has been a Professor of signal processing and Director of Tampere International Center for Signal Processing, leading a group of about 60 scientists. From 2001 to 2006, he was nominated Academy Professor by Academy of Finland. His research interests include signal processing, coding theory, spectral techniques, and statistics.

EURASIP Journal on Applied Signal Processing 2004:1, 92–107

Microarray BASICA: Background Adjustment, Segmentation, Image Compression and Analysis of Microarray Images

Jianping Hua
Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA
Email: huajp@ee.tamu.edu

Zhongmin Liu
Advanced Digital Imaging Research, 2450 South Shore Boulevard, Suite 305, League City, TX 77573, USA
Email: liuzm@adires.com

Zixiang Xiong
Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA
Email: zx@lena.tamu.edu

Qiang Wu
Advanced Digital Imaging Research, 2450 South Shore Boulevard, Suite 305, League City, TX 77573, USA
Email: qwu@adires.com

Kenneth R. Castleman
Advanced Digital Imaging Research, 2450 South Shore Boulevard, Suite 305, League City, TX 77573, USA
Email: castleman@adires.com

Received 14 March 2003; Revised 23 September 2003

This paper presents microarray BASICA: an integrated image processing tool for background adjustment, segmentation, image compression, and analysis of cDNA microarray images. BASICA uses a fast Mann-Whitney test-based algorithm to segment cDNA microarray images and performs postprocessing to eliminate the segmentation irregularities. The segmentation results, along with the foreground and background intensities obtained with the background adjustment, are then used for independent compression of the foreground and background. We introduce a new distortion measurement for cDNA microarray image compression and devise a coding scheme by modifying the embedded block coding with optimized truncation (EBCOT) algorithm (Taubman, 2000) to achieve optimal rate-distortion performance in lossy coding while still maintaining outstanding lossless compression performance. Experimental results show that the bit rate required to ensure sufficiently accurate gene expression measurement varies and depends on the quality of cDNA microarray images. For homogeneously hybridized cDNA microarray images, BASICA is able to provide from a bit rate as low as 5 bpp the gene expression data that are 99% in agreement with those of the original 32 bpp images.

Keywords and phrases: microarray BASICA, segmentation, Mann-Whitney test, lossy-to-lossless compression, EBCOT.

1. INTRODUCTION

The cDNA microarray technology is a hybridization-based process that can quantitatively characterize the relative abundance of gene transcripts [1, 2]. Contrary to conventional methods, microarray technology promises to monitor the transcript production of thousands of genes or even the whole genome simultaneously. It thus provides a new and powerful enabling tool for genetic research and drug discovery. To produce cDNA microarrays, the mRNA of the control and test samples are first reverse-transcribed into cDNA and fluorescently labeled with different dyes (typically red and green). Then the fluorescent targets are mixed and allowed to hybridize with gene-specific cDNA clones printed in an array format on a glass microslide. Finally, by scanning the microslide with a laser and capturing the photons emitted from different dyes into different channels with a confocal fluorescence microscope, a two-channel 16-bit microarray

image is obtained in which the pixel intensities reflect the level of mRNA expression. Usually a microarray image is shown RGB composite format, where the red and green channels correspond to the two channels of the microarray image obtained while the blue channel is set to zero. With the help of signal processing and data analysis operations such as ratio statistics, classification, and genetic regulatory network design, microarray images can shed light on the possible regulation rules of transcription production sought by biologists and clinicians.

Microarray images cannot be used for genetic data analysis directly. Appropriate image processing procedures are to be performed in order to extract information from the images for downstream analysis. Thousands of cDNA target sites must first be identified as the foreground by an image segmentation algorithm. Then the intensity pair (R, G) that represents gene expression levels of both channels is extracted from every foreground target site with appropriate background adjustment. Subsequent data analysis is normally conducted based on the log ratio $\log R/G$ of the intensity pair. As the very first step of cDNA microarray signal processing, the accuracy of image processing is critical to the reliability of subsequent data analysis. Many image processing schemes have been developed for this purpose in recent years and can be found in various commercial and noncommercial software packages [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. Generally, because each channel of the microarray image is typically more than 15 MB in size, highly efficient compression is necessary for data backup and communication purposes. In order to save storage space and alleviate the transmission burden for data sharing, the search for good progressive compression schemes that provide sufficiently accurate genetic information for data analysis at low bit rates while still ensuring good lossless compression performance has become the focus of cDNA microarray image compression research recently [3, 4, 20].

This paper introduces a new integrated system called microarray BASICA. BASICA brings together the image processing procedures required to accomplish the aforementioned information extraction and data analysis, including background adjustment, segmentation, and compression. A fast Mann-Whitney test-based algorithm is presented for the initial segmentation of cDNA microarray images. This new algorithm can save up to 50 times the number of repetitions required from the original algorithm [5]. The resulting images are then postprocessed to remove the segmentation irregularities. The segmentation results, along with the foreground and background intensities, are saved into a header file for both data analysis and compression. A novel distortion measure is introduced to evaluate the accuracy of extracted information. Based on this measure and the information provided by the header file, a new image compression scheme is designed by modifying the embedded block coding with optimized truncation (EBCOT) algorithm [3], which is now incorporated in the JPEG2000 standard. Our experiments show that there appears to be no common bit rate that ensures sufficiently accurate gene expression data for different cDNA microarray images. On cDNA microarray images of good quality, BASICA is able to provide from a bit rate as low as 5 bpp (bit per pixel) the gene expression data that are 99% in agreement with those of the original 32 bpp images.

2. DETAILS OF MICROARRAY BASICA

Microarray BASICA provides solutions to both processing and compression of cDNA microarray images. The major components of BASICA and their relationship with the elements of a microarray experiment are shown in Figure 1. Each two-channel microarray image acquired through the laser scanner is first sent to the *segmentation* component, where the target sites are identified. With the result of segmentation, the *background adjustment* component estimates each spot's foreground and background intensities and calculates the log ratio values based on the background-subtracted intensities. After this, the calculated log ratio values along with the segmentation information and other necessary data related to each spot are output for downstream data analysis. In the mean time, BASICA compiles the segmentation result and extracted intensities into a header file. With this header file, the *compression* component encodes the foreground and background of both channels of the original image into progressive bitstreams separately. The generated bitstreams, plus the header file, are saved into a data archive for future access or are transmitted as shared data. On the other hand, to utilize the archived or transmitted data, BASICA can either quickly retrieve the necessary genetic information saved in the header file or reconstruct the microarray image with available bitstreams through the *reconstruction* component and redo the segmentation and background adjustment.

2.1. Segmentation with postprocessing

Segmentation is performed to identify the target sites in each spot where the hybridization occurs. In [8], various existing segmentation schemes are summarized and categorized into four groups: (1) fixed circle segmentation, (2) adaptive circle segmentation, (3) adaptive shape segmentation, and (4) histogram segmentation.

Although the shape of a target site is determined by the physical attributes of the DNA probes and the mechanism of the printing procedure, most target sites are round or donut-like in shape. The fixed circle segmentation, which sets a round region of constant diameter in the middle of each spot as the target site, appears to be the most straightforward method and is provided in most existing software packages [9, 11, 12, 13, 17, 18]. The radius of the foreground is set either by a default value as a parameter of the robot arrayer and laser scanner or empirically determined by the user. The fixed circle method runs fast and performs well when the microarray spots are perfectly hybridized and aligned. In practical cases, however, the spots are far from perfect due to unpredictable nonuniform hybridization across the spot or misalignment of the probe array. GenePix [13] uses the adaptive circle segmentation to accommodate the varying sizes of different target sites, and Dapple [11] finds the best matched

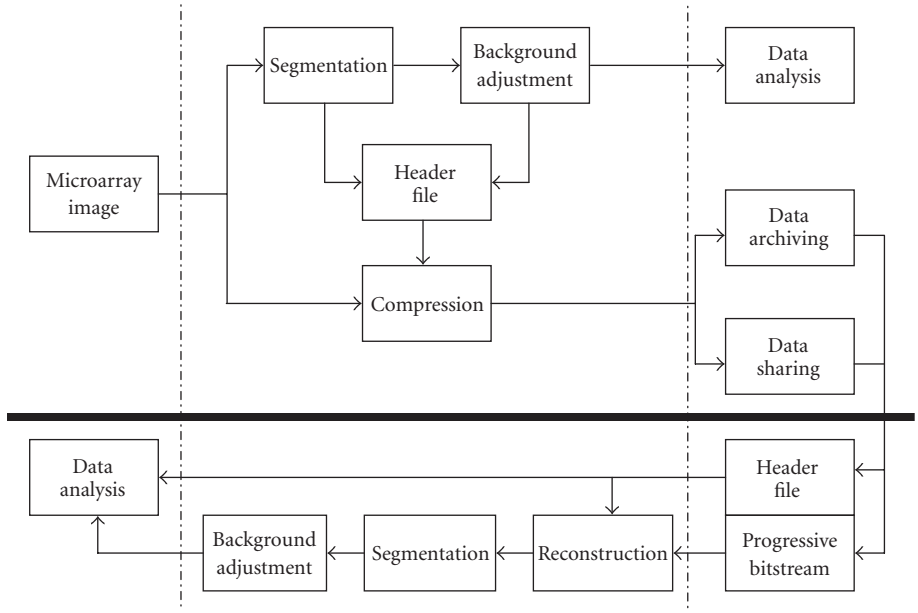


Figure 1: The major units of BASICA.

position of the round region in each spot to cope with the misalignment.

Neither the fixed nor the adaptive circle segmentation can accommodate the variances in shape of the target sites in the images. To tackle this problem, more accurate and sophisticated segmentation methods are needed. The segmentation technique introduced in [8] uses *seeded region growing* [21], while other methods [5, 6, 10, 15, 17, 19] rely on more conventional histogram-based segmentation algorithms. The histogram-based methods generally compute a histogram of pixel intensities for each spot. Methods in [10, 17, 19] adopt a percentile-based approach, which sets the pixels in a high percentile range of the histogram as the foreground and those in a low range as the background. Methods in [6, 15] use a threshold-based approach. To ensure correct segmentation, methods in [10, 15] employ repetitions to find the most stable segmentation. The histogram-based segmentation demonstrates good performance when a target site has a high hybridization rate, that is, a high intensity. However, the intensities of most target sites are actually very close to the local background intensities, and it is hard to segment correctly by finding a threshold based on the histogram only. In an attempt to solve this problem, Chen et al. introduced a Mann-Whitey test-based segmentation method in [5].

So far, no single segmentation algorithm can meet the demands of all microarray images. Segmentation algorithms are normally designed to perform well on microarray images acquired by certain type of arrayers and scanners. It is therefore hard to compare them directly.

2.1.1. Mann-Whitney test-based segmentation

In BASICA, we use the Mann-Whitney test-based segmentation algorithm introduced by Chen et al. in [5]. The Mann-Whitney test is a distribution-free rank-based two-sample test, which can be applied to various intensity distributions caused by irregular hybridization processes that are difficult to handle by conventional thresholding methods. Here we first give a brief description of the Mann-Whitney test-based segmentation algorithm.

Consider two independent sample sets X and Y. Samples $X_1, X_2, \ldots, X_m$ are randomly selected from set X, and $Y_1, Y_2, \ldots, Y_n$ are randomly selected from set Y. All $N = m + n$ samples are sorted and ranked. Denote R_i as the rank of the ith sample, $R(X_i)$ as the rank of sample X_i, and $R(Y_i)$ as the rank of Y_i. These ranks are used to test the following hypotheses:

(H_0) $P(X < Y) \geq 0.5$,
(H_1) $P(X < Y) < 0.5$.

Define the rank sum of the m samples from X as

$$T = \sum_{i=1}^{m} R(X_i). \tag{1}$$

To avoid deviations caused by ties, T is commonly normalized as

$$\overline{T} = \frac{T - m((N+1)/2)}{\sqrt{nm/N(N-1)\sum_{i=1}^{N} R_i^2 - nm(N+1)^2/4(N-1)}}. \tag{2}$$

Hypothesis (H_0) will be rejected if $\overline{T}$ is greater than a certain quantile $w_{1-\alpha}$, where α is the significance level.

In microarray image segmentation, hypothesis (H_1) corresponds to the case that set X is the high-intensity foreground and set Y is the low-intensity background, and

hypothesis (H_0) corresponds to the reverse case. To segment a target spot, a predefined target mask (obtained by selecting, unifying, and thresholding strong targets) is first applied to the spot. Pixels inside the mask correspond to set X, and pixels outside correspond to set Y. To start the test, n samples are randomly selected from set Y, while m samples with lowest intensities are selected from set X. If hypothesis (H_0) is accepted, the pixel with lowest intensity is removed from set X, and m sample pixels are reselected. The test is repeated until hypothesis (H_0) is rejected. Then the pixels left in set X are considered as the foreground at significance level α. The foregrounds obtained from the two channels are united into one to produce the final segmentation result.

The repetitive nature of this algorithm makes it cumbersome for real-time implementation. So, in BASICA, we proposed a fast Mann-Whitney test-based algorithm [3] which runs much faster while generating identical segmentation results.

2.1.2. Speeding up Mann-Whitey test-based segmentation algorithm

Assume that the predefined target mask is obtained according to the way described in [5, 6]. Samples $X_1, X_2, \ldots, X_m$ and $Y_1, Y_2, \ldots, Y_n$ are picked from the foreground and background, respectively. Without loss of generality, it suffices to assume that $X_1 \le X_2 \le \cdots \le X_m$ and $Y_1 \le Y_2 \le \cdots \le Y_n$. Since $X_1, X_2, \ldots, X_m$ are the smallest m samples in set X, all other samples can be determined if X_1 is set. Then Mann-Whitney test-based segmentation is actually an optimization problem of minimizing X_1 subject to $\overline{T} \ge w_{1-\alpha}$. Chen et al.'s approach takes a large number of repetitions to reach the final segmentation. However, it turns out that the number of repetitions can be significantly reduced by carefully choosing the starting point and search strategy.

BASICA first finds an upper bound of the optimal X_1, denoted by $X_1^{\max}$, which is related to $Y_1, Y_2, \ldots, Y_n$. With (2), $\overline{T} \ge w_{1-\alpha}$ can be written as

$$\sum_{i=1}^{m} R(X_i) \ge w_{1-\alpha} \sqrt{\frac{nm}{N(N-1)} \sum_{i=1}^{N} R_i^2 - \frac{nm(N+1)^2}{4(N-1)}} + m\frac{N+1}{2}. \tag{3}$$

In the right-hand side of (3), only $\sum_{i=1}^{N} R_i^2$ is associated with X_1. If no tie exists, the ranks are from 1 to N and the sum is $\sum_{i=1}^{N} i^2$. If there is a tie, the ranks of the tied samples are the average of those ranks if there would have been no tie, and induce a reduction on the sum. A property of this reduction is that it is only related to the number of samples tied at that value. If there are k samples having the same value, the deduction is $(1/12)(k^3 - k)$. With this property, one can easily reduce the upper bound of $\sum_{i=1}^{N} R_i^2$. Assume that ΔY is the decrease in the sum caused by the ties in the sorted $Y_1, Y_2, \ldots, Y_n$, then we have

$$\sum_{i=1}^{N} R_i^2 \le \sum_{i=1}^{N} i^2 - \Delta Y, \tag{4}$$

where the equation holds when $X_1, X_2, \ldots, X_m$ have no tie among themselves and share no tie with any sample in $Y_1, Y_2, \ldots, Y_n$. In most cases, the difference is very small and the bound is quite tight.

To simplify the notation, we use $\sigma_{\max}$ to denote

$$\sqrt{\frac{nm}{N(N-1)} \left(\sum_{i=1}^{N} i^2 - \Delta Y \right) - \frac{nm(N+1)^2}{4(N-1)}}. \tag{5}$$

Then $X_1^{\max}$ must satisfy the inequality

$$\sum_{i=1}^{m} R(X_i) \ge w_{1-\alpha}\sigma_{\max} + m\frac{N+1}{2}, \tag{6}$$

no matter what $X_2, X_3, \ldots, X_m$ can be for as long as the assumption $X_1 \le X_2 \le \cdots \le X_m$ holds. So, to find $X_1^{\max}$ is to find the smallest X_1 so that the smallest rank sum of $X_1 \le X_2 \le \cdots \le X_m$ still satisfies inequality (6). To associate $X_1^{\max}$ with known information $Y_1, Y_2, \ldots, Y_n$, assume that $Y_u < X_1^{\max}$. Then the minimum rank sum is $\sum_{i=1}^{m} R(X_i) = \sum_{i=1}^{m} (u+i)$, when $Y_u < X_1 \le X_2 \le \cdots \le X_m < Y_{u+1}$. By solving the inequality with $\sum_{i=1}^{m} R(X_i) = \sum_{i=1}^{m} (u+i)$, u can be obtained as

$$u = \left\lceil \frac{w_{1-\alpha}\sigma_{\max}}{m} + \frac{n}{2} \right\rceil. \tag{7}$$

Thus, the upper bound $X_1^{\max}$ is the smallest sample in X that is larger than Y_u. For any sample set $X_1, X_2, \ldots, X_m$ with $X_1 \ge X_1^{\max}$, hypothesis (H_0) can be rejected outright. Since $X_1, X_2, \ldots, X_m$ normally have similar intensities which bring on consecutive ranks, $X_1^{\max}$ is very close to the actual threshold. Hence, the repetitions can be greatly reduced if backward repetitions based on $X_1^{\max}$ are applied.

Besides changing the starting point and repetition direction, a two-tier repetition strategy can be used to reduce the repetition in case the upper bound is not so tight as expected. In the first tier, one does not perform the repetition in a pixel-by-pixel manner, but in a leaping manner instead. Then a pixel-by-pixel repetition follows up and locates the exact segmentation in the second tier. Larger step size means fewer repetitions in the first tier but more in the second, while smaller step size has the opposite effect. A natural choice of the repetition steps is indicated by $Y_1, Y_2, \ldots, Y_n$ when n is not very large. The whole algorithm is described as follows.

Step 1: Calculate u using (7).

Step 2: Find the smallest m samples from set X that are larger than Y_u, and execute the Mann-Whitney test.

Step 3: If hypothesis (H_0) is rejected, then set $u = u - 1$ and go to step 2, otherwise, go to step 4.

Step 4: $u = u + 1$. Find the smallest m samples from set X that are larger than Y_u, and begin the pixel-by-pixel repetition in backward manner.

It should be noted that this modified Mann-Whitney test-based segmentation algorithm may not always generate identical results with Chen et al.'s original algorithm. In order

Table 1: The comparisons of the number of repetitions between Chen et al.'s algorithm and our modified method used in BASICA at different significance levels.

α	0.001	0.005	0.01	0.05
Chen et al.	328.7	269.1	270.9	226.3
BASICA	7.5	7.3	5.9	3.7

to obtain identical results, the backward-searching nature of the new algorithm requires the normalized rank sum in (2) to be strictly increasing during the repetition of the original algorithm. This is not guaranteed due to the occurrences of ties in the sorted samples. In one extreme case, when all N samples have the same intensity, the devisor will become zero and the normalized rank sum will be infinity. Actually, Chen et al.'s original algorithm can be viewed as trying to find the largest foreground that rejects hypothesis (H_0), while the modified algorithm in BASICA tries to find the smallest foreground that accepts the hypothesis H_0. Since in most cases the normalized rank sum will be strictly increasing, we expect the segmentation results of the modified algorithm to be identical to the original algorithm most of the time.

The comparisons of the number of required repetitions between Chen et al.'s algorithm and our modified algorithm are given in Table 1. Results are averaged over 504 spots in both channels from different test images. Both algorithms set $m = n = 8$ and use the same randomly selected samples from the predefined background for the Mann-Whitney test. We find that the segmentation results on all test spots of the sample images used in this study are identical between the original algorithm and the modified algorithm. From the table, we observe that the modified algorithm reduces the number of repetitions by up to 50 times from what is required of the original algorithm.

2.1.3. Postprocessing

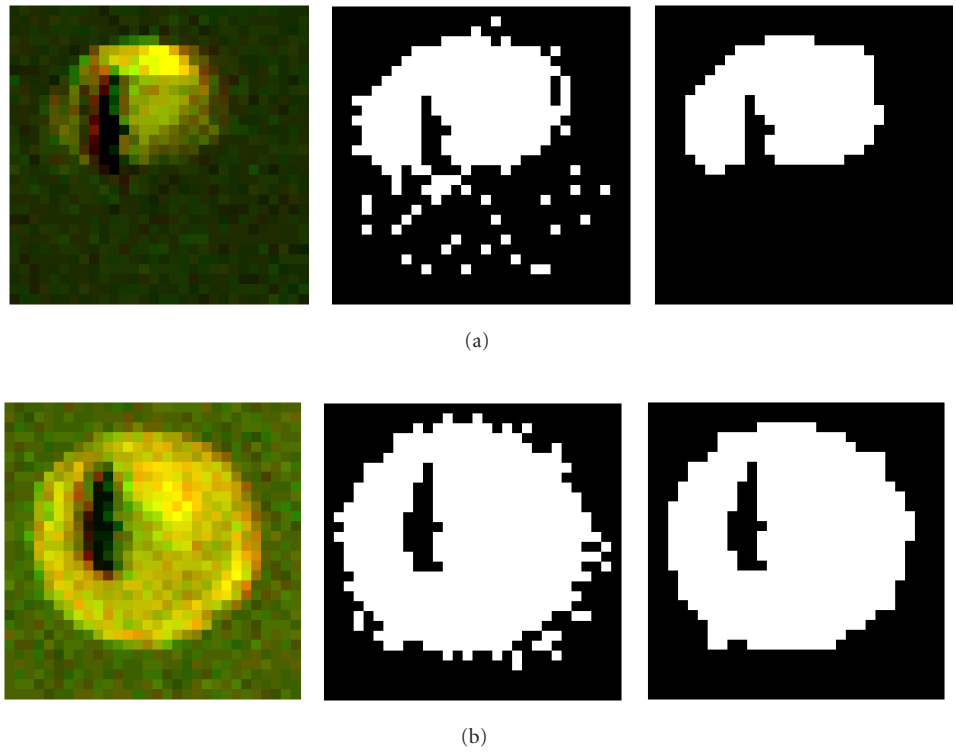
Like common threshold-based segmentation algorithms, there are always many annoying shape irregularities in the segmentation results obtained by the Mann-Whitney test-based algorithms. These irregularities occur randomly and can severely reduce the compression efficiency. Thus, an appropriate postprocessing procedure is necessary to achieve efficient compression. Moreover, because most irregularities are pixels with a high probability of noise corruption, eliminating them is unlikely to compromise the accuracy of subsequent data extraction and analysis.

In BASICA, we categorize possible irregularities into two types and employ different methods to eliminate them. The first type includes isolated noisy pixels or tiny regions, which can be observed from the lower half of the segmentation result in Figure 2a. These irregularities are caused usually by nonspecific hybridization or undesired binding of fluorescent dyes to the glass surface. The second type includes the small branches attached to the large consolidated foreground regions, which are visible in the segmentation results of Figure 2. Since these irregularities are located between the foreground and background, their intensities are also inbetween, making them vulnerable to noise corruption. The irregularities in most segmentation results are usually made up of both these two types. For the first type, BASICA will detect and remove them directly from the foreground. As for the second type, BASICA applies an operation similar to the standard morphological pruning [22]. By removing and pruning repetitively, BASICA can successfully eliminate most irregularities in three to five repetitions. The right column of Figure 2 shows the postprocessing results on the original segmentation which are to be used for the compression of the images. Figure 3 shows a portion of a microarray image and its segmentation results.

2.2. Background adjustment

It is commonly believed that the pixel intensity of the foreground reflects the joint effects of the fluorescence and the glass surface. To obtain the expression level accurately, the intensity bias caused by the glass surface should be estimated and subtracted from the foreground intensity, and this process is known as background adjustment. Since there is no hybridization in the background area, the background intensity is normally measured and treated as an intensity bias. Although mean pixel intensity has been adopted in almost all existing schemes as the foreground intensity, several methods have been developed for background intensity estimation. The major differences of various methods lie in two aspects: (1) on which pixels the estimation is based and (2) how to calculate the estimation. Regarding the first aspect, the regions chosen for background estimation vary from a global background to a local background. For the global background, the background regions in all spots are considered, and a global background intensity is estimated and subtracted from every foreground intensity [9, 16]. The global background ignores possible variance between subarrays and spots. So, in [9], partial global background estimation is performed based on the background of one subarray or on several manually selected spots. The more common approach is to estimate the background intensity based on the local background for each target site separately. The local background can be the entire background region in one spot [18], or, to avoid interference from the foreground, it can be the region with a certain distance from the foreground target site [7, 11, 13, 15, 17]. In the extreme case, the algorithm in [14] uses the pixels on the border of each spot as the local background. However, using too few pixels increases the possibility of a large variance in background estimation. As to the second aspect, almost all existing systems adopt mean or median to measure the expression level. Besides these, mode and minimum are also used in some softwares [6, 16]. Unlike all the methods mentioned above, a morphological opening operation is performed in [8] to smooth the whole background and then estimate the background by sampling at the center of the spot.

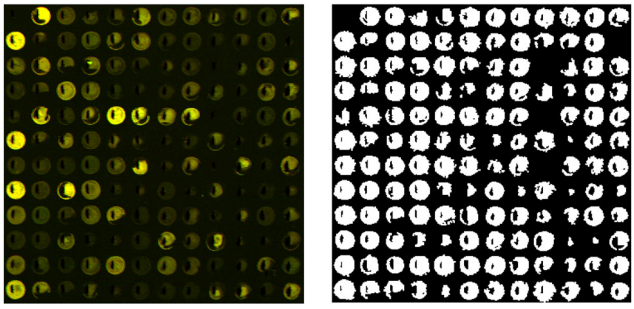
Some commercial software packages [9, 16] offer more than one choice for background adjustment. ArrayMetrix [9] provides up to nine methods, while ArrayVision [10]



(a)

(b)

Figure 2: Segmentation and postprocessing of two typical spots. The left column shows the original microarray spots in RGB composite format. Some intensity adjustments are applied in order to show them clearly. The middle column shows the corresponding segmentation results using the Mann-Whitney test with significance level $\alpha = 0.001$. The right column shows the final segmentation results after postprocessing.



(a) (b)

Figure 3: (a) Part of a typical cDNA microarray image in RGB composite format. Some intensity adjustments were applied in order to show the image clearly. (b) The segmentation results of (a).

provides seven ways of background region determination and six choices of averaging method. Experiments in [8] show that different background adjustment methods have significant impact on the log ratio values subsequently obtained. However, there is no known criterion to measure which approach is more accurate than the others.

BASICA chooses the average of pixel intensities in the local background as the estimate of background intensity. To prevent possible biases caused by either the higher intensity values of the pixels adjacent to the foreground target sites or the lower intensity values of the dark hole regions in the middle of the spots, the local background used in BASICA is the background defined by the predefined target mask obtained through the segmentation.

2.3. Data analysis

Because so many elements impact the pixel intensities of the microarray image, genetic researchers do not use the absolute intensities of the two channels, but the ratio between them to measure the relative abundance of gene transcription. Not all genetic information extracted are reliable enough for data analysis. If the spot has so poor quality that no reliable information can be extracted, it is qualified as a false spot; otherwise, it is a valid spot. For a valid spot k, the expression ratio is denoted by

$$T_k = \frac{R_k}{G_k} = \frac{\mu_{FR_k} - \mu_{BR_k}}{\mu_{FG_k} - \mu_{BG_k}}, \tag{8}$$

where R_k and G_k are the background-subtracted mean intensities of the red and green channels, respectively, μ_{FR_k} and μ_{FG_k} are the respective foreground mean intensities, and μ_{BR_k}

and μ_{BG_k} are the respective estimated background mean intensities. Because expression ratio has an unsymmetric distribution, which contradicts the basic assumptions of most statistic tests, the log ratio $\log T_k = \log R_k/G_k$ is commonly used instead in most applications. In addition to the log ratio, an auxiliary measure which is often helpful in data analysis is the log product $\log R_k G_k$. However, since the log transform does not have constant variance at different expression levels, some alternative transforms like $g\log$ [23] have recently been introduced. In gene expression studies, such transformed ratios are ordinarily normalized and quantized into three classes: down-regulated, up-regulated, and invariant. Expression level extraction and quantization provide the starting point for subsequent high-level data analysis, and their accuracy is crucially important. Therefore, compression schemes should be designed to minimize the distortion in the image, and their performance should be assessed by agreement/disagreement in gene expression level measurement caused by the compression. These topics will be discussed in detail in Sections 2.4 and 3.3.

2.4. Image compression

Since microarray images contain huge amounts of data and are usually stored at the resolution of 16 bpp, a two-channel microarray image is typically between 32 and 64 MB in size. Efficient compression methods are highly desired to accommodate the rapid growth of microarray images and to reduce the storage and transmission costs. Currently, the common method to archive microarray images is to store them losslessly in TIFF format with LZW compression [24]. However, such an approach does not exploit 2D correlation of data between pixels and does not support lossy compression. Due to the huge data size, microarray images require efficient compression algorithms which support not only lossless compression but also lossy compression with graceful degradation of image quality for downstream data analysis at low bit rates.

Recently, a new method known as the segmented LOCO (SLOCO) was introduced in [20]. This method exploits the possibility of lossy-to-lossless compression for microarray images. SLOCO is based on the LOCO-I algorithm [25], which has been incorporated in the lossless/near-lossless compression standard of JPEG-LS. SLOCO employs a two-tier coding structure. It first encodes microarray images lossily with near-lossless compression, then applies bit-plane coding to the quantization error to refine the coding results until lossless compression is achieved. SLOCO can generate a partially progressive bitstream with a minimum bit rate determined by the compression of the first tier, and the coding is conducted on the foreground and background separately.

In BASICA, we also incorporate lossy-to-lossless compression of microarray images. The aims of compression in BASICA are twofold: (1) to generate progressive bitstreams that can fulfill the requirements of signal processing and data analysis at low bit rates for data sharing and transmission applications and (2) to deliver competitive lossless compression performance to data archiving applications with a progressive bitstream. To achieve these objectives, the compression scheme in BASICA treats the foreground and background of microarray images separately. Obviously, the foreground and background usually have significant intensity differences and they are relatively homogeneous in their corresponding local regions. Hence, by compressing the foreground and background separately, the compression efficiency is expected to improve significantly. This is done by utilizing the outcomes of segmentation. Before encoding, BASICA saves all necessary segmentation information into a header file for subsequent compression.

SLOCO in [20] is based on spatial domain predictive coding. In contrast, BASICA employs bit-plane coding in the transform domain. Bit-plane coding enables BASICA to achieve truly progressive bitstream coding at any rate. To allow lossy compression, an appropriate distortion measurement is needed. Generally, medical image compression requires visually imperceptible differences between the lossily reconstructed image and the original. Traditional distortion measures, such as mean square error (MSE), are poor indicators for this purpose. However, unlike other types of medical images, the performance of microarray image compression does not depend on visual quality judgement, but instead on the accuracy of final data analysis. Therefore, it is reasonable to adopt a distortion measure adherent to the requirements of data analysis. Since almost all existing data analysis methods use the transformed expression values, we should seek to minimize the distortion under these measurements. In BASICA, we adopt distortion measures based on the log ratios and the log products because they are the most used transforms in common applications. However, as we will see later, the scheme employed in BASICA can be easily adapted for other transform measures.

The log ratios and the log products decouple the data of two channels into two separate log intensities, $\log R$ and $\log G$. This ensures that the compression can be done on each channel independently. Without loss of generality, we only refer to the R channel in the rest of the paper.

BASICA currently employs the MSE of $\log R$ as the distortion measurement, which is defined as

$$\mathrm{MSE}_{\log R} = \frac{1}{N}\sum_{i=1}^{N}\left(\log R_i - \log \hat{R}_i\right)^2, \tag{9}$$

where N is the total number of spots in the microarray image, and R_i and $\hat{R}_i$ are background-subtracted mean intensities obtained from spot i of the original and reconstructed image, respectively.

There is a direct relationship between the MSE of log intensity and the traditional MSE. For spot k, its log intensity $\log R_k$ can be further written as

$$\log R_k = \log\left(\mu_{FR_k} - \mu_{BR_k}\right) = \log\left(\frac{1}{M_k}\sum_{i=1}^{M_k} X_i - \mu_{BR_k}\right), \tag{10}$$

where M_k is the total number of pixels in the foreground of spot k, and X_i is the intensity of the ith pixel. So the unit error

$\Delta \log R_k$ is associated with the unit error ΔX_j of jth pixel by

$$\Delta \log R_k = \frac{\Delta X_j}{M_k(\mu_{FR_k} - \mu_{BR_k})}. \quad (11)$$

For the pixels in the background, because most existing schemes do not compute the average intensity as μ_{BR_k} but use nonlinear operations such as modulo or median filtering, the above derivation no longer holds. The foreground and background pixels have different impacts on the log intensity and should be considered separately.

Equation (11) indicates that the MSE of log intensity is actually a weighted version of traditional MSE. The weight $1/M_k(\mu_{FR_k} - \mu_{BR_k})$ is a constant for pixels in the same spot and is inversely proportional to the spot's intensity and foreground size. The higher a spot's intensity or foreground size, the larger its allowable reconstruction error.

Quite similarly, one can easily derive other MSE distortion measurements for other transforms. For example, the g log transform in [23] is

$$g(R_k) = \log\left(\mu_{FR_k} - \alpha + \sqrt{(\mu_{FR_k} - \alpha)^2 + c}\right), \quad (12)$$

where α and c are parameters estimated from the microarray image. Then, with straightforward derivation, one can associate the unit error $\Delta g(R_k)$ with the unit error ΔX_j of jth pixel by

$$\Delta g(R_k) = \frac{\Delta X_j}{M_k\sqrt{(\mu_{FR_k} - \alpha)^2 + c}}. \quad (13)$$

Thus, the MSE of g log is also a weighted version of traditional MSE, and like MSE of log ratio, the measurement allows larger distortions in spots of high intensities.

Although we can derive different distortion measurements for different transforms, the compression scheme in BASICA can only be designed based on one type of distortion measurement. As mentioned before, in BASICA we choose MSE of log ratio as the distortion measurement.

With the help of (11), we introduce a new lossy-to-lossless compression scheme in BASICA by modifying EBCOT [26] with several techniques specifically designed for the requirements of microarray technology. First, to encode the foreground and background separately, we modify the EBCOT to compress arbitrarily shaped regions. Then we apply intensity shifts and bit shifts on the coefficients to minimize the MSE of log intensity.

EBCOT, which is a state-of-the-art compression algorithm incorporated in JPEG2000 standard, offers a fully progressive bitstream of excellent compression efficiency with plenty of useful functionalities. In EBCOT, a 2D integer wavelet transform is applied for lossy-to-lossless image compression. Block-based bit plane coding is used to generate the bitstream of each subband. To achieve the optimal rate-distortion performance, the coding procedure consists of three passes in each bit plane using three context modeling primitives. The bitstreams of all subbands are multiplexed into a layered one via a fast bisectional search for the given target bit rate.

2.4.1. Modifying EBCOT for microarray image coding

Our major modifications to EBCOT are the following.

Header file. A header file is necessary for saving the information which will be used in the encoding and decoding procedures. To ensure that the encoder and decoder can correctly compress and reconstruct the foreground and background independently, the segmentation information must be saved in the header file. Besides, (11) indicates that the mean intensities of the foreground and background are also needed by the compression algorithm. To save storage memory, these data are coded with LZW compression. Although the segmentation information and spot intensities are enough for the compression component, other data, such as variances of pixel intensities in each spot, can also be saved in the header file for quick genetic information retrieval. In the practical implementation, the header file will be generated before encoding and must be transmitted and decoded first.

Shape-adaptive integer wavelet transform. Like other frequency-domain-based coding schemes, in BASICA the transform is performed before bit-plane coding during the encoding phase and after the bit-plane reconstruction during the decoding phase. To ensure lossless compression, integer wavelet transforms are required. The wavelet transforms are conducted on the foreground and background independently to prevent any interference between the coefficients from adjacent areas. Since the segmented foreground and background always have irregular shapes, critically sampled integer wavelet transforms for arbitrarily shaped objects are needed to ensure coding efficiency. Many approaches have been proposed for 2D shape-adaptive wavelet transforms. Our proposed coding scheme uses odd-symmetric extensions over object boundaries described in [27].

Object-based EBCOT. After shape-adaptive integer wavelet transform, we modify the EBCOT context modeling for arbitrarily shaped regions. The extension of EBCOT algorithm to shape-adaptive coding is rather straightforward. Because the shape-adaptive integer wavelet transform is critically sampled, the number of wavelet coefficients is the same as those in the original regions. Using the wavelet-domain shape mask, one can easily tell whether a coefficient belongs to a region to be coded. If any neighbor of that coefficient falls outside the region, we just set that neighboring coefficient's values to zero, thus making it insignificant in context modeling. We call the resulting coder object-based EBCOT.

Intensity shifts. To minimize the initial MSE, the average intensity of the image is subtracted from each pixel before encoding and added back after decoding. Unlike eight-bit natural images, the foreground of a microarray image normally has an exponential intensity distribution. The exponential distribution property of the foreground makes the global average intensity subtraction less effective. However, the pixels

in the foreground of any spot k normally have similar intensities and roughly have a symmetric distribution around μ_{FR_k}. So, for the encoding of the foreground, each pixel in spot k is subtracted μ_{FR_k} instead of the global average intensity. Since μ_{FR_k} is already saved in the header file, intensity shifts do not cost any overhead. With intensity shifts, the distribution of foreground intensities are transformed into a symmetric shape with a high peak around zero. As for the background compression, through our experiments, we find that the pixels in the background actually have a roughly symmetric intensity distribution, suggesting that the global average intensity subtraction will be appropriate.

Bit shifts. EBCOT uses block-based bit-plane coding. In order to minimize the distortions at different rates, one must code the bit planes of different spots according to their impacts on the MSE of log intensity. One straightforward solution is to scale the coefficients of each spot with the spot's weight, so bits at the same bit plane of all spots have the same impacts on the MSE of log intensity. However, because the weights are noninteger fractions, lossless compression cannot be ensured under such a scaling. Furthermore, although one can round them to the closest integer as an approximation, any scaler w will increase a coefficient's information up to $\lceil \log_2 w \rceil$ bits, which can lead to a very poor lossless compression performance. In BASICA, we apply the scaling by bit shifts, which is a good approximation and meanwhile does not compromise the performance of lossless compression. For spot k, BASICA obtains

$$S_k = \left\lfloor \log_2 \left(M_k(\mu_{FR_k} - \mu_{BR_k})\right) + 0.5 \right\rfloor. \tag{14}$$

Let $S^{\max} = \max\{S_1, S_2, \ldots, S_N\}$. Then it scales the coefficients of spot k by upshifting them $S^{\max} - S_k$ bits.

Background compression. With careful consideration, bit shifts have not been applied in the background compression in BASICA for several reasons. First, since there exist different approaches to compute the background intensity, and the values obtained by these methods also vary a lot, it is unclear how to find a unique weight for each pixel like what BASICA has for foreground compression. Second, unlike isolated target sites in the foreground, the local background is normally connected to each other. Thus, bit shifts will bring abrupt intensity changes along the borders of spots, which will in turn lower the compression efficiency significantly in lossless coding performance. Even though one can figure out the weights through a formula similar to (11) based on certain background extraction methods, there will be a significant tradeoff on lossless compression, which is about 0.8 bpp according to our experiments. So, in BASICA, we apply a global average intensity subtraction and no bit shifts on the background compression, that is, the traditional MSE measure is used for rate-distortion optimization. Normally, the pixel intensities in the background are located in a very small range, which means that the background is pretty homogenous. Thus, compression with traditional MSE measure should be able to represent the background with fairly small bit rates.

To this end, the final code of a two-channel microarray image is composed of five different parts: a header file and two bitstreams representing the foreground and background, respectively, from each channel.

3. EXPERIMENTAL RESULTS AND DISCUSSION

Experiments have been conducted to test the performance of BASICA with eight microarray images from two different sources. We used three test images from the National Institutes of Health (NIH). Each of these images contains eight subarrays arranged in 2×4 format. In each subarray, the spots are arranged in a 29×29 format. There are a total of 20184 spots in all the three NIH images. In addition to these, we also tested on another set of five test images obtained from Spectral Genomics Inc. (SGI). Each of the SGI images contains eight subarrays arranged in 12×2 format, and in each subarray, the spots are arranged in a 16×6 format. These five SGI images contain a total of 9960 spots. The target sites in the NIH images exhibit noticeable irregular hybridization effect and have irregular brightness patterns across the spots. The intensities of these target sites span over a large range and vary considerably. The target sites in the SGI images appear to be hybridized more homogeneously, and many of them have nearly perfect circular shape.

In the experiments, for each two-channel image, the summed bit rate of all the bitstreams from both channels, plus the shape information, were reported in bpp format, which represents either the compression bit rate or the reconstruction bit rate, depending on the type of test performed. And the corresponding bit rate of the uncompressed original image is 32 bpp. BASICA first segmented the image and generated the header file. The average overhead of the header file was 0.5 bpp for the NIH images and 0.24 bpp for the SGI images, based on the postprocessed segmentation results. The header file overheads were smaller on the SGI images because of different settings of the microarray arrayers used to acquire the images: there were much fewer spots in each SGI image than those in each NIH image. After generating the header file, the foreground and background of each channel were compressed independently.

3.1. Comparisons of wavelet filters and decomposition levels

The framework of the proposed compression scheme in BASICA does not specify which wavelet filters and how many wavelet decomposition levels used. In order to find the optimal choice for microarray image compression, we compare the results generated with different wavelet filters and decomposition levels. All the results presented in this section are based on the NIH images unless stated otherwise.

Table 2 lists the lossless coding results by BASICA using nine different wavelet filters with one-level wavelet decomposition. From these we found that the compression results vary only in a small range of about 0.07 bpp. Among all the nine sets of filters, the 5/3 wavelet filters achieved the best result. This is probably because the 5/3 wavelet filters have relatively shorter filter lengths, and therefore fit better with the small sizes of the segmented regions. Nevertheless, as the

Table 2: Lossless compression results (in bpp) of BASICA using different integer wavelet filters with one-level wavelet decomposition. The results are averaged over the NIH images.

Wavelet filters	$9/7-F$	$(2+2,2)$	5/3	$S+P$	(4,2)	(2,4)	(4,4)	(6,2)	2/6
File size	13.99	14.01	13.97	14.03	14.01	13.97	13.99	14.04	14.00

Table 3: Lossless compression results (in bpp) of BASICA using the 5/3 wavelet filters with different wavelet decomposition levels. The results are averaged over the NIH images.

Decomposition levels	1 level	2 levels	3 levels	4 levels	5 levels
File size	13.97	14.00	14.01	14.02	14.02

discrepancies in the results were small, the choice of the wavelet filters appeared to be not critical to the system performance.

Table 3 lists the lossless coding results by BASICA with different wavelet decomposition levels. Only the best-performing 5/3 wavelet filters were evaluated in these tests. The performance appeared to get worse when the decomposition level increased and compression with only one-level decomposition achieved the best result. This is partly due to the fact that although with more decompositions more data energy is compacted into smaller subbands, it also introduces a higher model-adaptation cost to arithmetic coding in the newly generated subbands, which cancels out the gains. Similar to the comparison among the wavelet filters, the discrepancies of lossless compression performance using different decomposition levels are very small. To confirm this observation, lossy compression tests were also performed to compare the performances based on the choices of the wavelet decomposition level.

To evaluate the effect of lossy compression on data analysis, the test images were first reconstructed at a target rate. Then the reconstructed images were processed and genetic information (i.e., log ratio) was extracted and compared with the same information extracted from the original images. To ensure credibility of the comparisons, the Mann-Whitney test-based segmentation started with the same selection of random pixels in the predefined background in both the reconstructed image and the original image. The segmentation was conducted under three different significance levels $\alpha = 0.001, 0.01$, and 0.05. At each significance level, log ratios were extracted and distortions were computed. The distortions shown are the average distortions at the three significance levels over the three test images. Both the l_1 distortion and l_2 distortion (i.e., MSE) of log intensity were used as the error measures. Figure 4 shows the average reconstruction errors using BASICA at different bit rates with three different decomposition levels of the 5/3 wavelet transform. From this figure, we can see that one-level decomposition yielded a significantly better performance than the others. Based on the above lossless and lossy compression results, we decided to use the 5/3 wavelet filters with one-level wavelet decomposition as a default setting in BASICA.

3.2. Comparisons of lossless compression

We first compared the lossless compression performance of BASICA with three current standard coding schemes: TIFF, JPEG-LS, and JPEG2000. In the comparisons, TIFF, JPEG-LS, and JPEG2000 all compress a microarray image as a single region and no header file is added. To evaluate the improvement brought by the postprocessing in segmentation, along with the intensity and bit shifts in compression, we also performed the tests of BASICA without the intensity and bit shifts and without postprocessing, respectively (denoted by BASICA w/o PP and BASICA w/o shifts, respectively, in Figures 5, 6, and 7, and Table 4).

The coding results are shown in Table 4. The TIFF format, which is commonly used in existing microarray image archiving systems, produced the poorest results, about 4 bpp worse than all the other methods compared. JPEG-LS achieved the best performance on the NIH images. But like TIFF, it does not support lossy compression. The proposed BASICA turned out to be about 0.27 bpp worse than JPEG-LS on the NIH images and 0.12 bpp better on the SGI images. Besides, BASICA was significantly better than JPEG2000 with the savings of 0.48 bpp and 0.56 bpp on the NIH and SGI images, respectively. BASICA without intensity and bit shifts yielded almost the same performance as BASICA in lossless compression. On the other hand, one can see clearly that the irregularities in segmentation reduced compression efficiency substantially. Without postprocessing, the average size of a header file was 0.33 bpp larger than that of BASICA on the NIH images and 0.09 bpp larger on the SGI images, respectively. Thus, BASICA with postprocessing was preferred on all the test images.

3.3. Comparisons of lossy compression

During the experiments, we also compared the lossy compression results at different bit rates. Since TIFF and JPEG-LS do not support the lossy compression functionality, JPEG2000 was the only standard compression scheme compared in the experiments. Our comparisons were based on three different measurements.

3.3.1. Comparisons based on l_1 and l_2 distortions

We first compared the rate-distortion curves based on the l_1 and l_2 distortions of log intensity. Figure 5 shows the average reconstruction errors of these methods at different bit rates. We observe that, due to the effect of relatively more homogeneous hybridization, the distortion on the SGI images was uniformly smaller than the distortion on the NIH images. JPEG2000 produced surprisingly small l_1 distortion values at

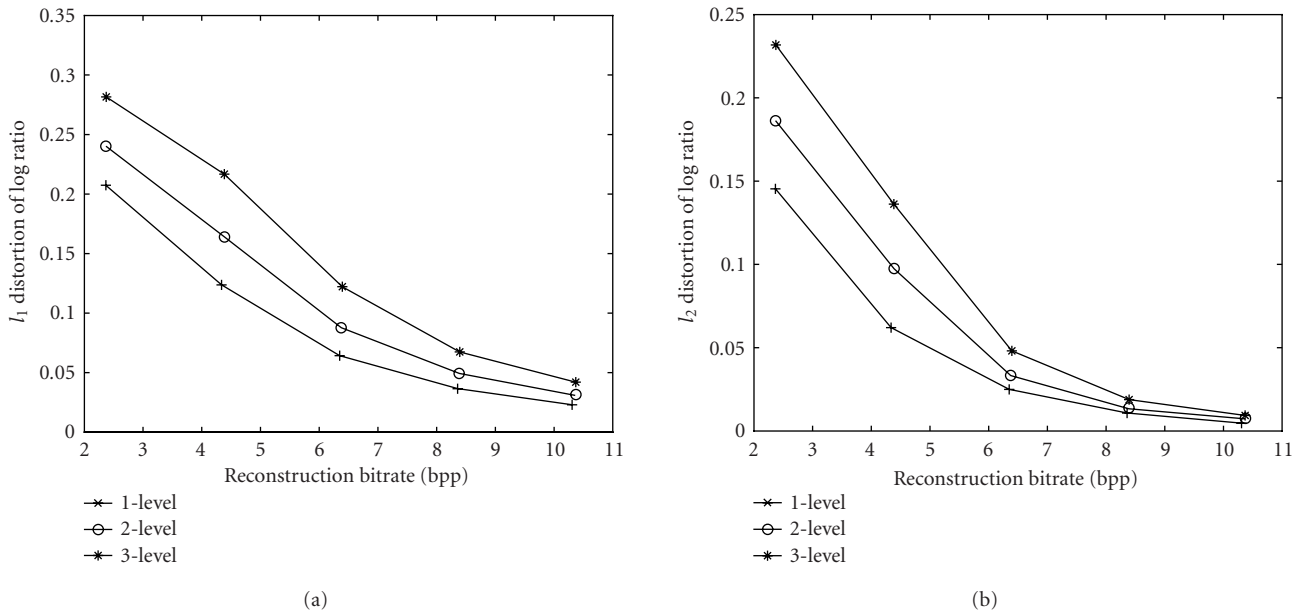


Figure 4: Rate-distortion curves of log ratio in terms of (a) l_1 distortion and (b) l_2 distortion with different wavelet decomposition levels at different reconstruction bit rates; 5/3 wavelet filters were used. The segmentation was performed at three different significance levels, $\alpha = 0.001, 0.01$, and 0.05, and three log ratios and their corresponding distortions were then obtained. The distortions shown are the averages of the three significance levels over the NIH images.

Table 4: Lossless compression results (in bpp) of different coding schemes.

Methods	TIFF	JPEG-LS	JPEG2000	BASICA w/o shifts	BASICA w/o PP	BASICA
Bit rates (NIH)	18.27	13.70	14.45	13.99	14.50	13.97
Bit rates (SGI)	17.21	14.49	14.93	14.31	14.46	14.37

low bit rates, only inferior to BASICA on the NIH images and similar to the others on the SGI images. Nevertheless, it produced relatively large l_2 distortion values. Apparently, without adjusting the MSE for log intensity, JPEG2000 spent too many bit rates on high-intensity pixels/spots, which led to high l_2 distortion. Furthermore, the distortion of JPEG2000 decayed slowly in both l_1 and l_2 senses. For bit rates beyond 6 bpp, it degraded to produce the worst distortion among all the methods. Without the intensity and bit shifts, BASICA performed poorly at lower bit rates. Only when the bit rates went above 6 bpp did its performance become acceptable. BASICA without postprocessing produced different performances on images of different sources. On the NIH images, it obviously suffered from the irregularities of segmentation, yielding a performance between BASICA and BASICA without the intensity, and bit shifts at low bit rates. But it quickly became worse than both of these schemes when the bit rates increased. On the SGI images, in which target sites had more uniform hybridization, there was almost no difference between its performance and BASICA's. Compared to the other schemes, BASICA yielded the best performance in both l_1 and l_2 distortions at all the bit rates on all test images.

3.3.2. Comparisons based on scatter plots

Besides l_1 and l_2 distortion measures, a more intuitively visual way to compare the distortion of different methods is by scatter plotting. Figure 6 shows the extracted log ratios and log products by different methods at a bit rate around 4 bpp for two test images. In each scatter plot, the blue diagonal line corresponds to the information extracted from the original images. From the plots, we can see that BASICA had a better performance than the other methods. BASICA without postprocessing had a worse performance on the NIH images and a good performance on the SGI images. JPEG2000 and BASICA without intensity and bit shifts yielded worse performances on both sets of test images. This observation is consistent with the results shown in Figure 5. Since a scatter plot cannot provide quantitative performance measurements and can only visually display the data for comparisons at one bit rate per plot, it does not provide a practical performance measurement.

3.3.3. Comparisons based on gene expression data

Rather than judging the performance based on the l_1 and l_2 distortion measures and the scatter plots, biologists and clinicians in gene expression studies are likely to care more

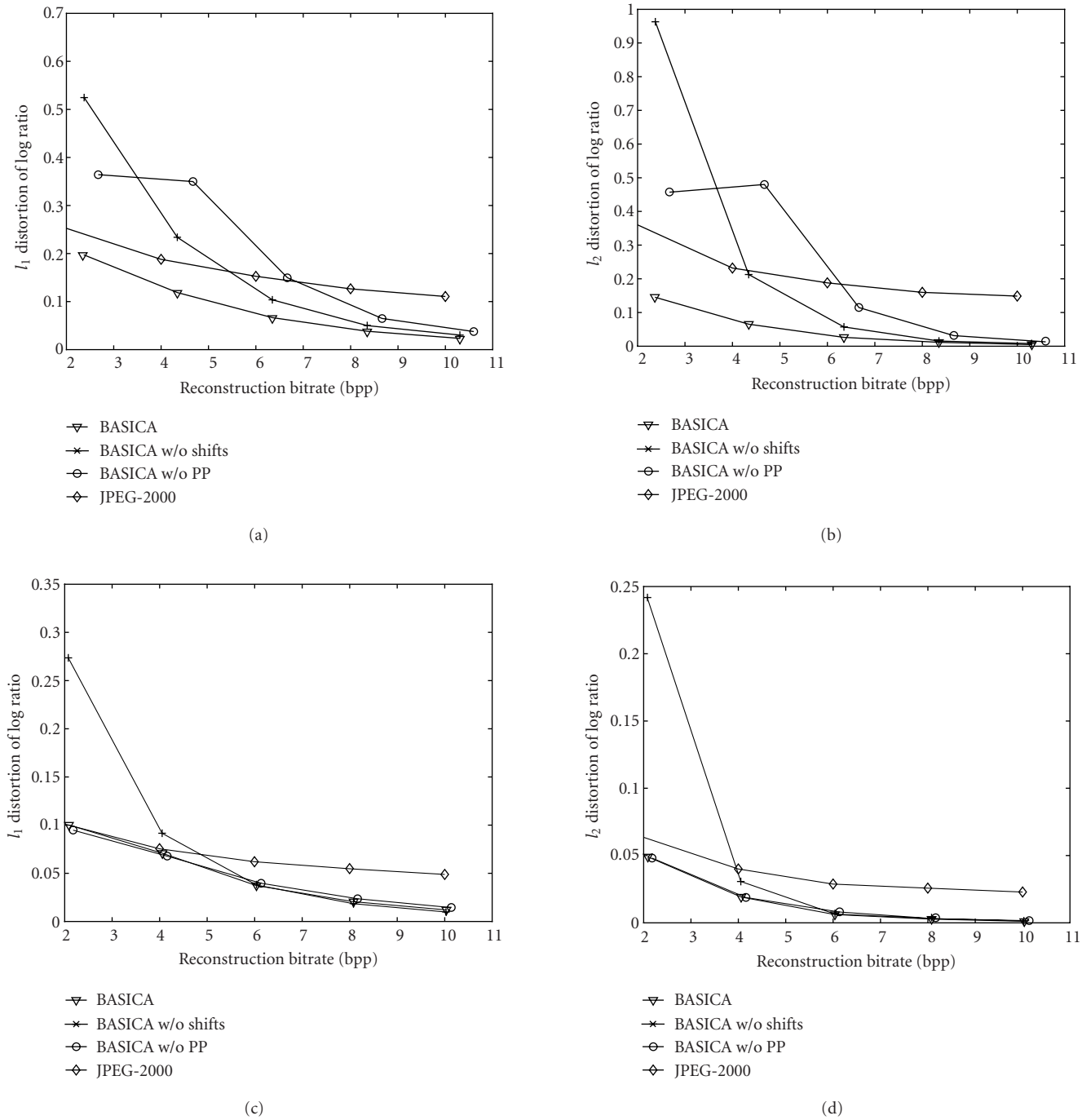


FIGURE 5: Rate-distortion curves of log ratio in terms of l_1 distortion (left column) and l_2 distortion (right column) under different reconstruction bit rates for different compression schemes: (a-b) results based on the NIH images; (c-d) results based on the SGI images. The segmentation was performed at the significance level $\alpha = 0.05$.

whether a gene is differently detected or identified due to a lossy compression. Hence, it is meaningful to look at the rate of disagreement on detection and identification between lossily reconstructed image and original image. The detection and identification disagreement are defined as follows.

(1) The detection disagreement is defined to be the valid spots in the original image being detected as false spot, or vice versa, after a lossy reconstruction.

(2) The identification disagreement is defined to be a different classification outcome among up-regulated, down-

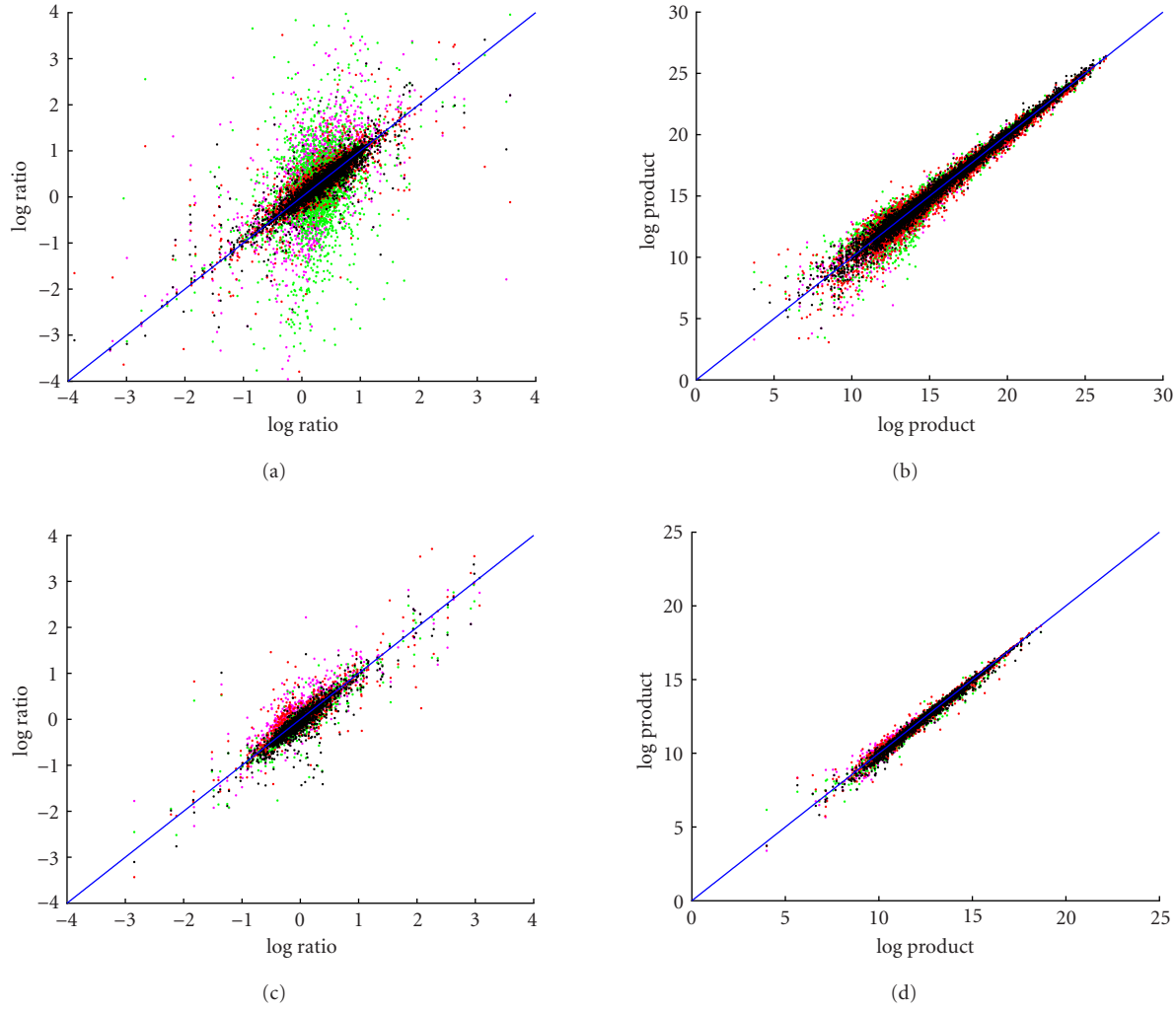


FIGURE 6: Scatter plots of log ratio (left column) and log product (right column) extracted from original images and reconstructed images using different schemes. (a-b) Results based on an NIH image: black: BASICA at 4.3 bpp; magenta: BASICA w/o shifts at 4.3 bpp; green: BASICA w/o PP at 4.7 bpp; red: JPEG2000 at 4.0 bpp. (c-d) Results based on an SGI image: black: BASICA at 4.1 bpp; magenta: BASICA w/o shifts at 4.1 bpp; green: BASICA w/o PP at 4.2 bpp; red: JPEG2000 at 4.0 bpp. The significance level in the Mann-Whitney test is $\alpha = 0.05$.

regulated, and invariant gene expression levels after a lossy reconstruction, even though the detection outcome is the same.

We conducted experiments using a simple quantitative model of gene expression data analysis to compare different methods. We determined that a spot was false if its foreground intensity was less than its background intensity in either channel or no foreground target site was found by the segmentation. We also decided that if the log ratio was larger or smaller than a certain threshold range $[\theta, -\theta]$, then the spot was up- or down-regulated; otherwise, it was invariant. For these experiments, no normalization was performed to reduce the interimage data variations. The experiments were performed on the NIH images and the SGI images separately and the results are shown in Figure 7. From this figure we can see that the identification disagreement rate was about 10 times higher than the detection disagreement rate. These results were similar to what have been shown in Figure 5. The disagreement caused by the lossy compression of JPEG2000 was comparable to that of BASICA only at 2 bpp, and dropped slowly when the bit rate increased. On the other hand, the disagreement caused by BASICA without intensity and bit shifts became acceptable only after 6 bpp. BASICA without postprocessing yielded a performance similar to that of BASICA on the SGI images but did worse on the NIH images. One can also observe that the disagreement rates on the NIH images were much higher than on the SGI images at the same bit rate. This is probably because NIH images are much noisier than SGI images, and hence require more bit rates to compress. These results are consistent with Figure 5, where the NIH images have much larger l_1 and l_2 distortions than the SGI images

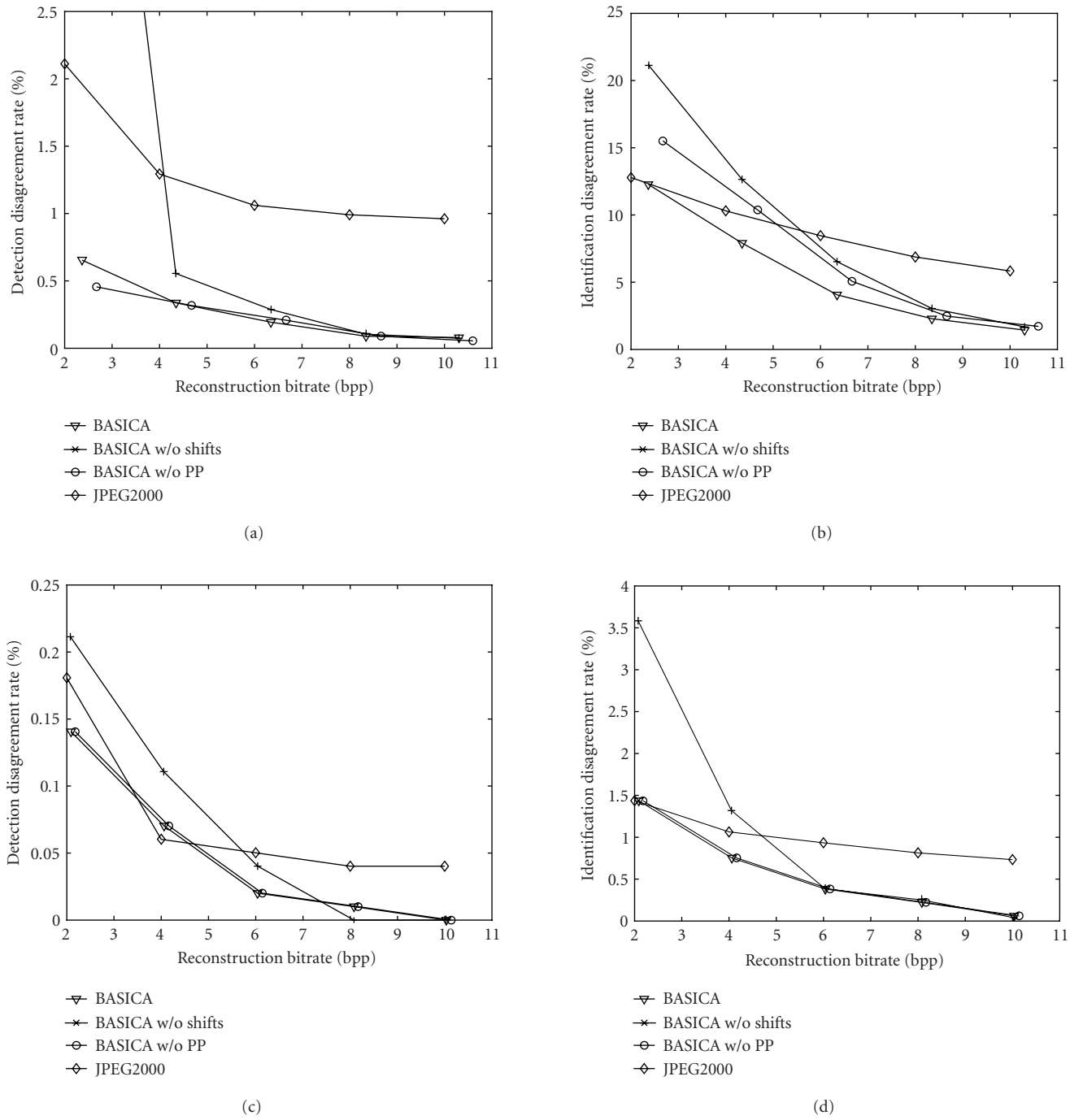


FIGURE 7: The disagreement rates versus the bit rates. The threshold parameters $\theta = 1$. The segmentation was performed at the significance level $\alpha = 0.05$. The left-column plots depict the detection disagreement rates versus the bit rates. The right-column plots depict the identification disagreement rates versus the bit rates. The disagreement rates shown are the averages of all images: (a-b) results based on the NIH images; (c-d) results based on the SGI images.

at the same bit rate. For the NIH images, the identification disagreement rate was larger than 10% at 2 bpp and was around 1.5% at 10 bpp. For the SGI images, the identification disagreement rate was smaller than 2.5% even at 2 bpp, and was around 0.1% at 10 bpp. All these results consistently suggested that one could hardly find a common bit rate that led to similar disagreement/agreement rates for different microarray images. For images with homogeneous

hybridization, which are becoming more available with the advance of microarray production technology, lossy compression at low bit rates appears to be viable for highly accurate gene expression data analysis.

4. CONCLUSIONS AND FUTURE RESEARCH

We have introduced a new integrated tool, microarray BASICA, for cDNA microarray data analysis. It integrates background adjustment, and image segmentation and compression in a coherent software system. The cDNA microarray images are segmented by a fast Mann-Whitney test-based algorithm. Postprocessing is performed to remove the segmentation irregularities. A highly efficient image coding scheme based on a modified EBCOT algorithm is presented, along with a new distortion measurement specially chosen for cDNA microarray data analysis. Experimental results show that cDNA microarray images of different quality require different bit rates to ensure sufficiently accurate gene expression data analysis. For homogeneously hybridized cDNA microarray images, BASICA is able to provide from a bit rate as low as 5 bpp the gene expression data that are 99% in agreement with those of the original 32 bpp images. Future research includes finding the optimal rate allocation between the background and foreground, and between the two channels of a cDNA microarray image.

ACKNOWLEDGMENTS

The authors would like to thank both Professor E. Dougherty and Dr. S. Shah for fruitful discussions and for separately providing the microarray test images used in this study. This work was supported by the NIH SBIR Grant 1R43CA94251-01, the NSF CAREER Grant MIP-00-96070, the NSF Grant CCR-01-04834, the ARO YIP Grant DAAD19-00-1-0509, and the ONR YIP Grant N00014-01-1-0531. This paper was presented in part at the Genomic Signal Processing and Statistics (GENSIPS), Raleigh, NC, October 2002.

REFERENCES

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–470, October 1995.

[2] *The Chipping Forecast*, *Nature Genetics*, vol. 21, suppl. 1, January 1999.

[3] J. Hua, Z. Xiong, Q. Wu, and K. Castleman, "Fast segmentation and lossy-to-lossless compression of DNA microarray images," in *Proc. Workshop on Genomic Signal Processing and Statistics*, Raleigh, NC, USA, October 2002.

[4] J. Hua, Z. Xiong, Q. Wu, and K. Castleman, "Microarray BASICA: Background adjustment, segmentation, image compression and analysis of microarray images," in *Proc. IEEE International Conference on Image Processing*, Barcelona, Spain, September 2003.

[5] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, no. 4, pp. 364–374, 1997.

[6] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, no. 9, pp. 1207–1215, 2002.

[7] Scanalytics, *MicroArray Suite For Macintosh Version 2.1 User's Guide*, August 2001.

[8] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed, "Comparison of methods for image analysis on cDNA microarray data," *Journal of Computational and Graphical Statistics*, vol. 11, no. 1, pp. 108–136, 2002.

[9] Raytest Isotopenmessgeraete GmbH, *AIDA Array Metrix User's Manual*, 2002.

[10] Imaging Research, *ArrayVision Version 7.0 Reference Manual*, 2002.

[11] J. Buhler, T. Ideker, and D. Haynor, "Dapple: Improved techniques for finding spots on DNA microarrays," Tech. Rep. 2000-08-05, Department of Computer Science and Engineering, University of Washington, Seattle, Wash, USA, 2000.

[12] A. J. Carlisle, V. V. Prabhu, A. Elkahtown, et al., "Development of a prostate cDNA microarray and statistical gene expression analysis package," *Molecular Carcinogenesis*, vol. 28, no. 1, pp. 12–22, 2000.

[13] Axon Instruments, *GenePix Pro 4.1 User's Guide and Toturial*, Rev. G., 2002.

[14] CLONDIAG chip technologies GmbH, *IconoClust 2.1 Manual*, 2002.

[15] X. Wang, S. Ghosh, and S. Guo, "Quantitative quality control in microarray image processing and data acquisition," *Nucleic Acids Research*, vol. 29, no. 15, pp. 75–82, 2001.

[16] Nonlinear Dynamics, *Phoretix Array version 3.0 User's Guide*, 2002.

[17] PerkinElmer Life Sciences, *QuantArray Analysis Software, Operator's Manual*, 1999.

[18] M. Eisen, *ScanAlyze User Manual*, 1999.

[19] A. N. Jain, T. A. Tokuyasu, A. M. Snijders, R. Segraves, D. G. Albertson, and D. Pinkel, "Fully automatic quantification of microarray image data," *Genome Research*, vol. 12, no. 2, pp. 325–332, 2002.

[20] R. Jornsten, W. Wang, B. Yu, and K. Ramchandran, "Microarray image compression: SLOCO and the effect of information loss," *Signal Processing*, vol. 83, no. 4, pp. 859–869, 2003.

[21] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[22] E. R. Dougherty, *An Introduction to Morphological Image Processing*, SPIE Optical Engineering Press, Bellingham, Wash, USA, 1992.

[23] B. P. Durbin, J. S. Hardin, D. M. Hawkins, and D. M. Rocke, "A variance-stabilizing transformation for gene-expression microarray data," *Bioinformatics*, vol. 18, suppl. 1, pp. S105–S110, 2002.

[24] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[25] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.

[26] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

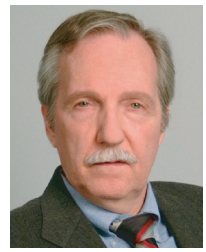
[27] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 725–743, 2000.

Jianping Hua received his B.S. and M.S. degrees in electrical engineering from the Tsinghua University, Beijing, China, in 1998 and 2000, respectively. Currently, he is pursuing his Ph.D. in electrical engineering at Texas A&M University, Tex, USA. His main research interests lie in image and video compression, joint source-channel coding, and genomic signal processing.

Zhongmin Liu received his B.S. degree in engineering physics and M.S. degree in reactor engineering from Tsinghua University, Tsinghua, China, in 1992 and 1994, respectively. He received the Ph.D. degree in electrical engineering from Texas A&M University in 2002. From 1994 to 1996, he was with the Precision Instrument Department, Tsinghua University, as a Research Assistant. From 1996 to 1998, he was with Hewlett-Packard China as an R&D Engineer. He is currently working in Advanced Digital Imaging Research. His research interests include DSP algorithm implementation, image and video processing, wavelet coding, joint source-channel coding, and pattern recognition.

Zixiang Xiong received his Ph.D. degree in electrical engineering in 1996 from the University of Illinois at Urbana-Champaign. From 1997 to 1999, he was with the University of Hawaii. Since 1999, he has been with the Department of Electrical Engineering at Texas A&M University, where he is an Associate Professor. He spent the summers of 1998 and 1999 at Microsoft Research, Redmond, Wash and the summers of 2000 and 2001 at Microsoft Research in Beijing. His current research interests are distributed source coding, joint source-channel coding, and genomic signal processing. Dr. Xiong received a National Science Foundation (NSF) Career Award in 1999, the United States Army Research Office (ARO) Young Investigator Award in 2000, and an Office of Naval Research (ONR) Young Investigator Award in 2001. He also received the Faculty Fellow Awards in 2001, 2002, and 2003 from Texas A&M University. He is currently an Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Signal Processing, and the IEEE Transactions on Image Processing.

Qiang Wu received his Ph.D. degree in electrical engineering in 1991 from the Catholic University of Leuven (Katholieke Universiteit Leuven), Belgium. In December 1991, he joined Perceptive Scientific Instruments Inc. (PSI), Houston, Texas, where he was a Senior Software Engineer from 1991 to 1994 and a Senior Research Engineer from 1995 to 2000. He was a key contributor to research and the development of the core technology for PSI's PowerGene cytogenetics automation products, including digital microscope imaging, image segmentation, enhancement, recognition, compression for automated chromosome karyotyping, and FISH image analysis for automated assessment of low-dose radiation damage to astronauts after NASA space missions. He is currently a Lead Research Engineer at Advanced Digital Imaging Research, Houston, Texas. He has served as Principal Investigator on numerous National Institutes of Health (NIH) SBIR grants. His research interests include pattern recognition, image processing, artificial intelligence, and their biomedical applications.



Kenneth R. Castleman received his B.S., M.S., and Ph.D. degrees in electrical engineering from The University of Texas at Austin in 1965, 1967, and 1969, respectively. From 1970 through 1985, he was a Senior Scientist at NASA's Jet Propulsion Laboratory in Pasadena, Calif, where he developed digital imaging techniques for a variety of medical applications. He also served as a Lecturer at Caltech and a Research Fellow at the University of Southern California (USC) and at the University of California, Los Angeles (UCLA). In 1984, he founded Perceptive Systems, Inc., a company that developed imaging workstations for cytogenetics. He is the author of more than 60 journal articles, two textbooks on digital image processing, and three patents. He has served as a Principal Investigator on more than a dozen government-sponsored research grants, and served on advisory boards of National Institutes of Health (NIH), the University of Texas, Carnegie-Mellon University, and the FBI. He is currently President of Advanced Digital Imaging Research, Houston, Texas. In 1994, Dr. Castleman was inducted into the Space Foundation Space Technology Hall of Fame. He is a Fellow of the American Institute for Medical and Biological Engineering and a past Chairman of the IEEE student branch at The University of Texas at Austin.

EURASIP Journal on Applied Signal Processing 2004:1, 108–114

A Digital Signal Processing Method for Gene Prediction with Improved Noise Suppression

Trevor W. Fox

Research and Development Department, Intelligent Engines Corporation, 903 42 St. SW, Calgary, Alberta, Canada T3C-1Y9
Email: tfox@bm.net

Alex Carreira

Department of Electrical and Computer Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4
Email: aycarrei@shaw.ca

Received 1 March 2003; Revised 15 September 2003

It has been observed that the protein-coding regions of DNA sequences exhibit period-three behaviour, which can be exploited to predict the location of coding regions within genes. Previously, discrete Fourier transform (DFT) and digital filter-based methods have been used for the identification of coding regions. However, these methods do not significantly suppress the noncoding regions in the DNA spectrum at $2\pi/3$. Consequently, a noncoding region may inadvertently be identified as a coding region. This paper introduces a new technique (a single digital filter operation followed by a quadratic window operation) that suppresses nearly all of the noncoding regions. The proposed method therefore improves the likelihood of correctly identifying coding regions in such genes.

Keywords and phrases: gene prediction, digital filter, DNA.

1. INTRODUCTION

Finding coding regions (exons) in a DNA strand involves searching amongst the many nucleotides that comprise a DNA strand. Typically a DNA molecule contains millions to hundreds of millions of elements [1]. The problem of finding exons in a DNA sequence is well suited to computers because DNA sequences can be represented by data that is easily processed by a computer. DNA strands can be represented by sequences of letters from a four-character alphabet. Convention dictates the use of the letters A, T, C, and G in each element to represent each of the four distinct nucleotides [1]. A nucleotide has two distinct ends: a 3′ end and a 5′ end. A covalent chemical bond links the 5′ end of one nucleotide to the 3′ end of another nucleotide. A DNA strand is comprised of many nucleotides linked in this fashion [1]. The DNA sequence representing a DNA strand consists of the letters A, T, C, and G listed in a left-to-right fashion corresponding to the nucleotides that make up the strand arranged left to right from their 5′ to 3′ ends [1].

A DNA strand can be divided into genes and intergenic spaces. Genes are responsible for protein synthesis. A gene can be further subdivided into exons and introns for cells with a nucleus (eukaryotes) [2]. Cells without a nucleus are called prokaryotes and do not contain introns [2]. The exons, coding regions within genes, are denoted by start and stop codons. Codons are a subsequence of three letters within the DNA sequence. Because codons are comprised of three letters from the four-letter alphabet that makes up a DNA sequence, there are 64 possible codons [1]. Of the 64 possible codons, there are one start codon and three stop codons, and the remainder of the codons correspond to one of the twenty possible amino acids of a protein [1]. The relationship between DNA sequences, genes, intergenic spaces, exons, introns, and codons is illustrated in Figure 1.

Some exons within the protein-coding regions of DNA sequences of eukaryotes tend to exhibit a period-three pattern [2, 3, 4, 5]. The period-three pattern of the exons can be exploited to predict gene locations and even predict specific exons within the genes of eukaryotic cells [2, 3, 4, 5].

Previous digital signal processing (DSP) methods for the identification of coding regions (exons) in DNA sequences include the application of the discrete Fourier transform (DFT) on overlapping windows [1, 3, 4] and the application of bandpass digital filters that are centered at $2\pi/3$ [2, 6]. The output of a bandpass digital filter centered at $2\pi/3$ can be thought of as one measure of the DNA spectral content at frequency $2\pi/3$. Digital filter methods are of interest because

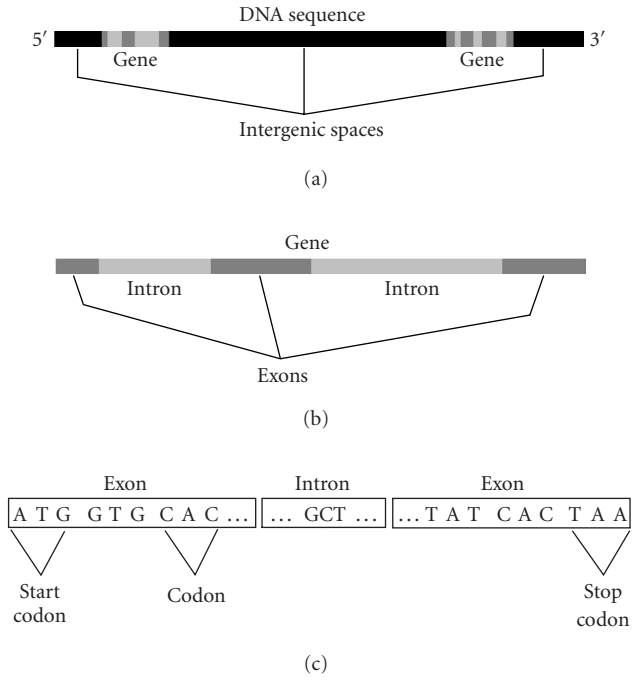


Figure 1: (a) An abstraction to illustrate the genes and intergenic spaces which comprise a DNA sequence. (b) An abstraction of a gene to illustrate the subdivision of a gene into exons and introns. (c) Various subsequences that comprise exons and introns in a gene (each three-letter grouping is a codon). The start codon is always ATG. However, one of the three possible stop codons is illustrated as (TAA).

they are significantly faster than the DFT method and they can be used to suppress more of the DNA background noise than it is possible by using the DFT method [2, 6].

DSP methods that only exploit period-three behaviour have many shortcomings. These methods are unable to reliably locate coding regions that do not have strong period-three characteristics. Methods based on hidden Markov models [7, 8, 9] provide superior results in these circumstances. The models used in these methods are also sufficiently accurate to account for exon and intron length distributions [10]. Alternatively, computational methods that exploit the heterogeneous statistical properties of DNA sequences to recursively segment homogeneous subsequences from their heterogeneous supersequences can be used for the identification of the borders between coding and noncoding regions [11, 12, 13]. The accuracy of these segmentation methods for coding region identification in DNA sequences surpasses the method presented in this paper and other DSP methods when applied to DNA sequences that do not have coding regions exhibiting a periodicity of three.

The method presented in this paper is an extension of DSP methods that exploit period-three behaviour. Previous DSP methods that exploit period-three behaviour do not entirely suppress the noncoding regions in the DNA spectrum at $2\pi/3$. As a result, a noncoding region may be incorrectly identified as a coding region. Also the methods presented in [2, 6] require four digital filter operations. In contrast, this paper presents a method that requires only one digital filter operation followed by a quadratic windowing operation. The quadratic window produces a signal that has almost zero energy in the noncoding regions. The proposed method can therefore improve the likelihood of correctly identifying coding regions over previous digital filtering methods. However, the accuracy of the proposed method suffers when dealing with coding regions that do not exhibit strong period-three behaviour. Also the methods presented in [7, 8, 9] are able to accurately model structures in genes, whereas the proposed method cannot. Despite these limitations, the method proposed in this paper can be used to generate one of the signals of a more complex gene finding method.

This paper is organized as follows. Section 2 reviews previous DSP methods for the identification of coding regions in DNA sequences. In particular, the DFT and digital filter methods are discussed. Section 3 presents a new computationally efficient one-step digital filter method for the identification of coding regions. Section 4 presents a new quadratic window operation that improves the suppression of noncoding regions from the DNA spectrum at frequency $2\pi/3$. In the example presented, noise suppression is improved by almost three orders of magnitude. Section 5 presents the conclusions of this research.

2. PREVIOUS DIGITAL SIGNAL PROCESSING METHODS FOR IDENTIFYING CODING REGIONS

Strands of DNA consist of four nucleotides (or bases), which are designated by the characters A, T, C, and G [1]. A character string composed of these four bases can be mapped to four signals [1]. The signal $u_A(n)$ takes the value of either 1 if A is present in the DNA sequence at index n, or 0 if A is absent at index n. For example, $u_A(n)$ for the DNA segment ATGCTGAA is 10000011. The signals $u_T(n)$, $u_C(n)$, and $u_G(n)$ can be obtained in a similar fashion.

The DFT of $u_A(n)$ over N samples is defined [14] as 3pt

$$U_A(k) = \sum_{n=0}^{N-1} u_A(n) e^{-j2\pi kn/N}, \quad 0 \le k \le N-1. \tag{1}$$

In a similar fashion, the DFT of $u_T(n)$, $u_C(n)$, and $u_G(n)$ can be obtained. For many genes, period-three behaviour has been observed and is useful for identifying coding regions [2, 3, 4, 5]. Specifically, the $(k = N/3)$-DFT coefficient magnitude is often significantly larger than the surrounding DFT coefficient magnitudes and corresponds to a coding region within the gene [1, 3, 4]. This effect varies and can be quite pronounced or quite weak, depending upon the gene [2].

A figure that can be used to measure the total spectral content of a DNA character string at frequency k is defined as [1, 4, 15]

$$S_{A+C+T+G}(k) = (U_A(k))^2 + (U_T(k))^2 + (U_C(k))^2 + (U_G(k))^2. \tag{2}$$

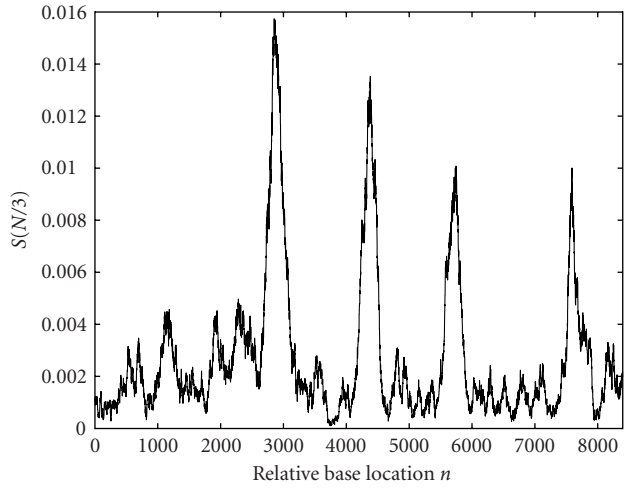


FIGURE 2: The signal $S_{A+C+T+G}(N/3)$ for gene F56F11.4 in the *C-elegans* chromosome III ($N = 351$).

The subscript of $S_{A+C+T+G}(k)$ indicates that all four nucleotide signals are considered. Corresponding to the previously described period-three behaviour, the value of $S_{A+C+T+G}(k)$ is large at $k = N/3$ when a coding region is present. The progression of $S_{A+C+T+G}(N/3)$ can be plotted by evaluating $S_{A+C+T+G}(N/3)$ over a window of N samples, sliding the window by one or more sample, and recalculating $S_{A+C+T+G}(N/3)$ [1]. This process can be carried out over the entire DNA sequence. As an example, consider the gene F56F11.4 in the *C-elegans* chromosome III. The value of $S_{A+C+T+G}(N/3)$ using $N = 351$ is plotted over the base numbers 7021 to 15080 in Figure 2.

The four dominant peaks in Figure 2 clearly indicate coding regions. However, a fifth coding region is present from 929 to 1135 but its small peak is obscured by $1/f$ DNA background noise. (The work presented in [15, 16, 17] observes the presence of $1/f$ background noise in DNA sequences.)

The DFT method for the identification of coding regions can be interpreted as a bandpass digital filter operation followed by a decimation operation [2]. The bandpass digital filter associated with the DFT method is centered at frequency $2\pi/3$ and has a minimum stopband attenuation of only 13 dB. High frequency selective bandpass digital filters for the identification of coding regions can be used instead of the DFT and have been presented in [2, 6] by Vaidyanathan and Yoon. The digital filter presented in [6] is a second-order antinotch filter. The digital filter presented in [2] is an eleventh-order bandpass digital filter with a minimum stopband attenuation of 60 dB.

The digital filter method for the identification of coding regions does not require the use of a sliding window [2, 6]. Instead, the signals $u_A(n)$, $u_C(n)$, $u_T(n)$, and $u_G(n)$ are individually processed using the same digital filter to produce the signals $y_A(n)$, $y_C(n)$, $y_T(n)$, and $y_G(n)$. A pseudomeasure of the total spectral content of a DNA sequence at frequency $2\pi/3$, $y_{A+C+T+G}(n)$, is given by [2, 6]

$$y_{A+C+T+G}(n) = |y_A(n)|^2 + |y_C(n)|^2 + |y_T(n)|^2 + |y_G(n)|^2. \quad (3)$$

The signal $y_{A+C+T+G}(n)$ produces large values in coding regions that exhibit strong period-three behaviour [2, 6] and is therefore an indicator for coding regions.

The digital filter method is much faster than the DFT method. For example, processing gene F56F11.4 in the *C-elegans* chromosome III using the DFT method requires 264 seconds on a 400 MHz Pentium II computer. In contrast, the digital filter method presented in [2] requires only 0.36 seconds, which is 733 times faster than the DFT method.

3. GENE PREDICTION USING A SINGLE DIGITAL FILTER

The methods presented by Vaidyanathan and Yoon in [2, 6] require a digital filtering operation for each of the four $u_A(n)$, $u_C(n)$, $u_T(n)$, and $u_G(n)$ signals for a total of four separate filtering operations. We now introduce a method that only requires one application of a digital filtering operation by filtering a single signal composed of $u_T(n)$ and $u_G(n)$. This new approach also removes much more of the DNA background noise than it is possible by using the methods presented in [2, 6]. In the following two sections, the optimization problem for creating this new signal is described and solved for a specific example.

3.1. Optimized signal construction

The number of digital filter operations can be reduced from four to one with the creation of a new signal that encapsulates the entire DNA sequence

$$u_{A+C+T+G}(n) = au_A(n) + cu_C(n) + tu_T(n) + gu_G(n), \quad (4)$$

where a, c, t, and g are real-valued parameters. Strand symmetry [18, 19, 20] can be exploited to further reduce the complexity of (4) to the sum of two terms. A long DNA sequence can be approximated using a two-symbol representation, where one symbol is either A or T and the other symbol is either C or G. In this case, the signal becomes

$$u_{T+G}(n) = tu_T(n) + gu_G(n). \quad (5)$$

Strand symmetry may not hold for shorter DNA sequences (on the order of 100 bases) and therefore strand symmetry should be verified before using (5) on short sequences. Section 3.2 compares the use of (4) and (5) for a test DNA sequence.

An optimization-based approach can be used to select the values of t and g (or a, c, t, and g if the strand symmetry is not used). A digital filter for gene prediction is first obtained from either the literature or from a suitable filter design method (this paper uses the digital filter presented in [2]). This digital filter is used in the optimization process to produce $v_{T+G}(n)$ from $u_{T+G}(n)$. A DNA sequence is selected where all of the coding regions are known. A pseudomeasure

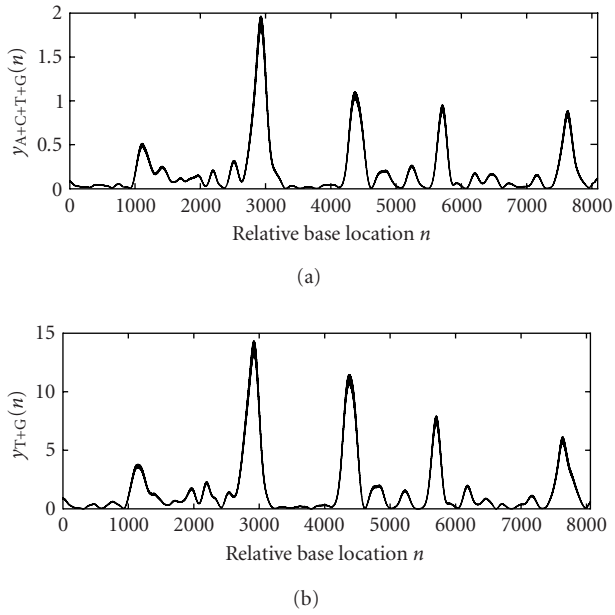


FIGURE 3: The signals $y_{T+G}(n)$ and $y_{A+C+T+G}(n)$ for gene F56F11.4 in the *C-elegans* chromosome III using the proposed single digital filter method.

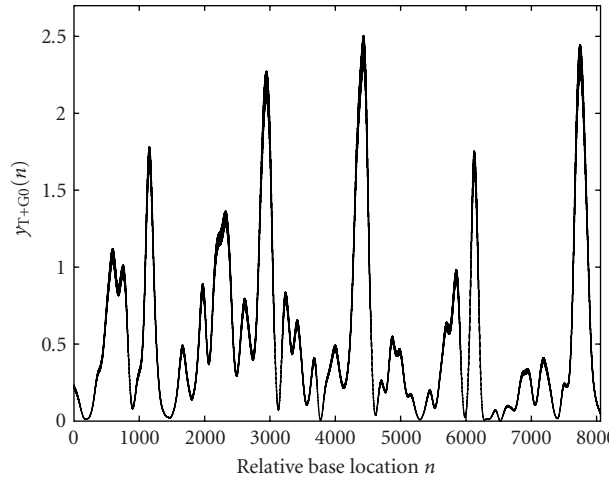


FIGURE 4: The signal $y_{T+G0}(n)$ for gene F56F11.4 in the *C-elegans* chromosome III.

of the total spectral content of a DNA sequence at $2\pi/3$ is given by

$$y_{T+G}(n) = v_{T+G}^2(n). \tag{6}$$

The ratio of $y_{T+G}^2(n)$ accumulated over all of the coding regions to $y_{T+G}^2(n)$ accumulated over all of the noncoding regions is maximized by choosing the t and g parameters:

$$\text{Maximize } \frac{\sum_{n_0 \in [\text{coding region}]} y_{T+G}^2(n_0)}{\sum_{n_1 \in [\text{noncoding region}]} y_{T+G}^2(n_1)}. \tag{7}$$

3.2. Applying the signal optimization

As an example, consider the use of the digital filter presented in [2] and the chromosome XVI of *S. cerevisiae* dataset. The quasi-Newton optimization method [21] is used to solve the above optimization problem for a two-symbol signal and for a four-symbol signal. The method proposed in this section is then used to process gene F56F11.4 in the *C-elegans* chromosome III over the base numbers 7021 to 15080 (see Figure 3). Figure 3 demonstrates that $y_{T+G}(n)$ and $y_{A+C+T+G}(n)$ are very similar due to the strand symmetry. The use of $y_{T+G}(n)$ is preferred because of its simplicity.

All five exons in Figure 3 are clearly visible in both $y_{T+G}(n)$ and $y_{A+C+T+G}(n)$. The remaining peaks do not have sufficient magnitude to obscure any of the coding regions. The total energy of $y_{T+G}(n)$ in the noncoding regions is defined as $\sum_{n \in [\text{noncoding region}]} {y_{T+G}}^2(n)$. This is a useful performance measure to gauge the effectiveness of a DSP gene prediction method for the suppression of the noncoding regions in $y_{T+G}(n)$. The total energy of $y_{T+G}(n)$ using the single digital filter method is 56.6. In contrast, the total energy of $y_{T+G}(n)$ in the noncoding regions using the multiple digital filter method as presented in [2] is 273.7, which is almost five times larger than the proposed single digital filter method. Clearly in this example, the proposed method improves the likelihood of correctly identifying the coding regions by reducing the total energy of $y_{T+G}(n)$ in the noncoding regions.

The initial coding region for gene F56F11.4 in the *C-elegans* chromosome III has a weak period-three characteristic, which is evident in Figures 2 and 3. In Figure 2, the initial coding region is obscured by noise. Optimizing the parameters t and g in $u_{T+G}(n)$ over a training sequence consisting of initial, internal, and terminal coding regions can be used to suppress a significant portion of this noise (see Figure 3). However, the relative height of the peak in $y_{T+G}(n)$ associated with the initial coding region is almost unchanged.

Our experiments indicate that the method proposed in this paper cannot be used to increase the relative height of the peaks in $y_{T+G}(n)$ associated with coding regions without also increasing the energy in the noncoding regions. We have attempted to optimize a new signal, $u_{T+G0}(n)$, that, when filtered, produces larger peaks for initial coding regions. A training dataset composed only of initial coding regions in XVI of *S. cerevisiae* was used to obtain t and g. Figure 4 shows $y_{T+G0}(n)$ for gene F56F11.4 in the *C-elegans* chromosome III. The relative height of the peak associated with the initial coding region shown in Figure 4 has increased but at the expense of a significant increase in the signal energy in the noncoding regions. Consequently, the use of $u_{T+G0}(n)$ has little practical benefit because the increased signal energy in the noncoding regions decrease the likelihood of correctly identifying the coding regions. Similar results can be obtained if t and g are optimized only for internal coding regions or only for terminal coding regions. In contrast, methods based on hidden Markov models [7, 8, 9] use sufficiently accurate models to predict the location of coding regions that do not have strong period-three characteristics.

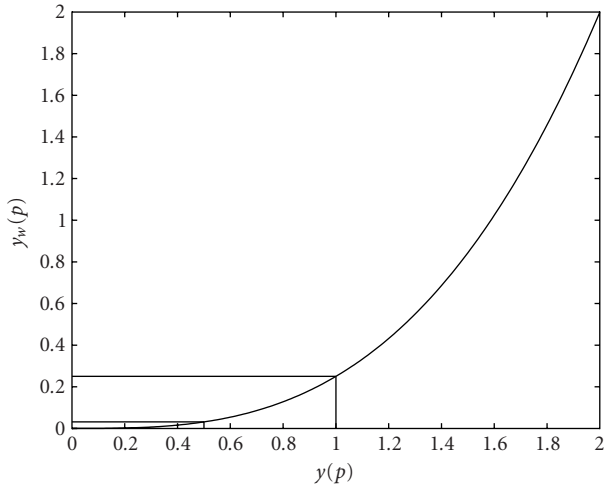


Figure 5: The quadratic window nonlinearity plotted for Maxvalue = 2.

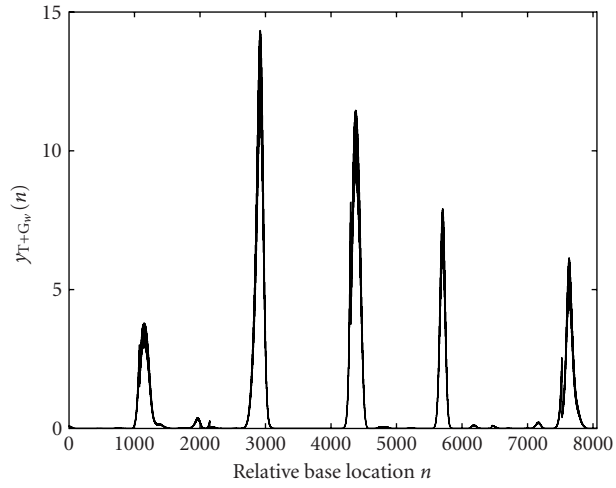


Figure 6: The signal $y_{T+G_w}(n)$ for gene F56F11.4 in the *C-elegans* chromosome III using the quadratic window (8).

4. A QUADRATIC WINDOW OPERATION TO SUPPRESS NONCODING REGIONS

The single digital filter method for the identification of coding regions does not always suppress all of the peaks found in the noncoding regions of $y_{T+G}(n)$ (see Figure 3). Consequently, the noncoding regions may obscure the coding regions in some datasets. To reduce uncertainty in the identification of coding regions, a new quadratic windowing operation is now introduced that can be used to effectively suppress the noncoding regions while preserving the coding regions. This quadratic windowing operation is performed after the single digital filter operation on $y_{T+G}(n)$.

The maximum value of $y_{T+G}(n)$ in a coding region is almost always greater than the maximum value of $y_{T+G}(n)$ in a noncoding region although the difference in magnitude between the two may be small. It is desirable to exaggerate the difference in magnitude between the coding and noncoding regions so that the coding regions can be more easily identified. To this end, a window of M samples is processed using the following operation:

$$y_w(p) = \left(\frac{y_{T+G}(p)}{\text{Maxvalue}}\right)^2 \cdot y_{T+G}(p), \quad 1 \le p \le M, \qquad (8)$$

where p is the window sample index, M is the number of samples in the window, $y_w(p)$ is the pth windowed sample value, and Maxvalue is the largest value of $y_{T+G}(p)$ in the window.

The quadratic windowing operation defined in (8) multiplies $y_{T+G}(p)$ by a value that approaches zero in a quadratic fashion as $y_{T+G}(p)$ approaches zero. Noncoding regions in the window that have sample values less than Maxvalue are effectively suppressed. Consider a window of samples that has maximum sample value of 2. The quadratic window operation produces $y_w(p)$ values of 0.0313 and 0.25 for $y_{T+G}(p)$ values that equal 0.5 and 1, respectively, as shown in Figure 5.

To preserve the coding regions in $y_{T+G}(n)$, the size of the window should not contain more than one coding region. In this case, the sole coding region in the window is not suppressed because the value of the largest sample, which belongs to the coding region, is not changed when using (8). A DNA sequence, where all of the coding regions are known, can be used to select the window size. The window size is set to a value less than the minimum number of samples between adjacent coding regions and greater than the number of samples of the widest coding region.

After a window of M samples has been processed, the window is then moved M samples, which prevents the successive windowing operations from overlapping.

The quadratic windowing operation is now applied to the gene F56F11.4 in the *C-elegans* chromosome III over the base numbers 7021 to 15080. Figure 3 shows the original $y_{T+G}(n)$ signal obtained using the method discussed in Section 3.2. The quadratic window of (8) is used to obtain the signal $y_w(p)$, as shown in Figure 6. The window size is set to $M = 1100$ samples. The five coding regions (exons) dominate the signal $y_w(n)$. In the coding regions, the signal $y_w(n)$ has been suppressed to near-zero values, which improves the certainty of correctly identifying the coding regions.

Table 1 compares the suppression of the noncoding regions by comparing the total energy in these regions for the multiple digital filter gene prediction method presented in [2], the single digital filter method presented in Section 3, and the single digital filter method followed by the quadratic window operation presented in this section. This numerical experiment used gene F56F11.4 in the *C-elegans* chromosome III over the base numbers 7021 to 15080.

The multiple digital filter method does not effectively minimize the total energy in the noncoding regions. The total energy in the noncoding regions for the multiple digital filter method is 720 times greater than the total energy in noncoding regions for the method proposed in this section and almost five times greater than the method presented in Section 3. As a result, a noncoding region may inadvertently

Table 1: A comparison of the performance between competing gene prediction methods.

Gene prediction method	Total energy in the noncoding regions
Single digital filter method followed by the quadratic window operation	0.38
Single digital filter method	56.6
Multiple digital filter method [2]	273.7

Table 2: A comparison of SNR values between competing gene prediction methods.

Gene	SNR (single digital filter method followed by the quadratic window operation)	SNR (multiple digital filter method [2])
F56F11.4	107	4
ZK250.9	225	18
ZK250.10	848	22
F54D8.1	64	11

be identified as a coding region when using the multiple digital filter method. In contrast, all five coding regions can easily be identified using the methods presented in this section.

The quadratic windowing method (single digital filter method followed by a quadratic window operation) is now compared in more depth with Vaidyanathan and Yoon's multiple digital filter method [2]. Table 2 compares the signal-to-noise ratio (SNR), see (9), for the following test genes: F56F11.4 in the *C-elegans* chromosome III, ZK250.9 and ZK250.10 in the *C-elegans* chromosome II, and F54D8.1 in the *C-elegans* chromosome III.

The SNR performance measure considers both the energy in the coding and noncoding regions. High SNR signals have low energy levels in the noncoding regions and high energy levels in the coding regions. For high SNR signals, the task of identifying coding regions is greatly simplified because the coding regions dominate over the noncoding regions

$$\mathrm{SNR} = \frac{\sum_{n_0 \in [\text{coding region}]} y_{\mathrm{T+G}}^2(n_0)}{\sum_{n_1 \in [\text{noncoding region}]} y_{\mathrm{T+G}}^2(n_1)}. \tag{9}$$

Table 2 shows that the multiple digital filter method consistently generates significant lower SNR signals than does the method proposed in this paper. Consequently, the task of identifying coding regions in signals generated by the multiple digital filter method is more problematic.

5. CONCLUSION

Methods for the identification of coding regions that solely rely on digital filters [2, 6] are unable to significantly attenuate the noncoding regions in $y_{\mathrm{T+G}}(n)$. Consequently, a noncoding region may inadvertently be identified as a coding region. This paper introduced a new DSP technique (a single digital filter operation followed by a quadratic window operation) that can be used to suppress nearly all of the noncoding regions in $y_{\mathrm{T+G}}(n)$. This paper demonstrated that the total energy in the noncoding regions of $y_{\mathrm{T+G}}(n)$ can be reduced by a factor of 720 compared to the previous digital filter techniques for gene F56F11.4 in the *C-elegans* chromosome III. As a result, the proposed method can improve the likelihood of correctly identifying coding regions.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their comments and valuable suggestions which helped in improving this paper.

REFERENCES

[1] D. Anastassiou, "Genomic signal processing," *IEEE Signal Processing Magazine*, vol. 18, no. 4, pp. 8–20, 2001.

[2] P. P. Vaidyanathan and B.-J. Yoon, "Digital filters for gene prediction applications," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, pp. 306–310, Pacific Grove, Calif, USA, November 2002.

[3] D. Anastassiou, "DSP in genomics," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 1053–1056, Salt Lake City, Utah, USA, May 2001.

[4] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya, and R. Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences," *Comput. Appl. Biosci.*, vol. 13, no. 3, pp. 263–270, 1997.

[5] J. W. Fickett, "Recognition of protein coding regions in DNA sequences," *Nucleic Acids Res.*, vol. 10, no. 17, pp. 5303–5318, 1982.

[6] P. P. Vaidyanathan and B.-J. Yoon, "Gene and exon prediction using allpass-based filters," in *Workshop on Genomic Signal Processing and Statistics*, Raleigh, NC, USA, October 2002.

[7] J. Henderson, S. Salzberg, and K. H. Fasman, "Finding genes in DNA with a hidden Markov model," *J. Comput Biol.*, vol. 4, no. 2, pp. 127–141, 1997.

[8] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman, "A generalized hidden Markov model for the recognition of human genes in DNA," in *Proc. of the 4th International Conference on Intelligent Systems for Molecular Biology*, Menlo Park, Calif, USA, 1996.

[9] A. Krogh, I. S. Mian, and D. Haussler, "A hidden Markov model that finds genes in E. coli DNA," *Nucleic Acids Res.*, vol. 22, no. 22, pp. 4768–4778, 1994.

[10] C. B. Burge and S. Karlin, "Finding the genes in genomic DNA," *Curr. Opin. Struct. Biol.*, vol. 8, no. 3, pp. 346–354, 1998.

[11] P. D. Cristea, "Large scale features in DNA genomic signals," *Signal Processing*, vol. 83, no. 4, pp. 871–888, 2003.

[12] W. Li, P. Bernaola-Galvan, F. Haghighi, and I. Grosse, "Applications of recursive segmentation to the analysis of DNA sequences," *Computers & Chemistry*, vol. 26, no. 5, pp. 491–510, 2002.

[13] W. Li, G. Stolovitzky, P. Bernaola-Galvan, and J. L. Oliver, "Compositional heterogeneity within, and uniformity between, DNA sequences of yeast chromosomes," *Genome Research*, vol. 8, no. 9, pp. 916–928, 1998.

[14] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[15] R. F. Voss, "Evolution of long-range fractal correlations and $1/f$ noise in DNA base sequences," *Phys. Rev. Lett.*, vol. 68, no. 25, pp. 3805–3808, 1992.
[16] W. Li, "The study of correlation structures of DNA sequences: a critical review," *Computers & Chemistry*, vol. 21, no. 4, pp. 257–271, 1997.
[17] W. Li and K. Kaneko, "Long-range correlation and partial $1/f^{\alpha}$ spectrum in a non-coding DNA sequence," *Europhys. Lett.*, vol. 17, no. 7, pp. 655–660, 1992.
[18] D. R. Forsdyke and J. R. Mortimer, "Chargaff's legacy," *Gene*, vol. 261, no. 1, pp. 127–137, 2000.
[19] W. Li, "The study of correlation structures of DNA sequences: a critical review," *Computers & Chemistry*, vol. 21, no. 4, pp. 257–272, 1997.
[20] J. W. Fickett, D. C. Torney, and D. R. Wolf, "Base compositional structure of genomes," *Genomics*, vol. 13, no. 4, pp. 1056–1064, 1992.
[21] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, Pa, USA, 1996.

Trevor W. Fox received his B.S. and Ph.D. degrees in electrical engineering from the University of Calgary in 1999 and 2002, respectively. Currently, he is working at the Intelligent Engines in Calgary, Canada. His main research interests include digital filter design, reconfigurable digital signal processing, and genomic signal processing.

Alex Carreira received his B.S. and M.S. degrees in electrical engineering from the University of Calgary, Canada, in 1999 and 2003, respectively. His main research interests are digital signal processing with programmable logic devices, configurable and reconfigurable computing, and rapid prototyping of systems for programmable logic devices.

EURASIP Journal on Applied Signal Processing 2004:1, 115–124

Gene Prediction Using Multinomial Probit Regression with Bayesian Gene Selection

Xiaobo Zhou
Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA
Email: zxb@ee.tamu.edu

Xiaodong Wang
Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
Email: wangx@ee.columbia.edu

Edward R. Dougherty
Department of Electrical Engineering, Texas A&M University, 3128 TAMU College Station, TX 77843-3128, USA
Department of Pathology, University of Texas MD Anderson Cancer Center, Houstan, TX 77030, USA
Email: e-dougherty@tamu.edu

Received 3 April 2003; Revised 1 September 2003

A critical issue for the construction of genetic regulatory networks is the identification of network topology from data. In the context of deterministic and probabilistic Boolean networks, as well as their extension to multilevel quantization, this issue is related to the more general problem of expression prediction in which we want to find small subsets of genes to be used as predictors of target genes. Given some maximum number of predictors to be used, a full search of all possible predictor sets is combinatorially prohibitive except for small predictors sets, and even then, may require supercomputing. Hence, suboptimal approaches to finding predictor sets and network topologies are desirable. This paper considers Bayesian variable selection for prediction using a multinomial probit regression model with data augmentation to turn the multinomial problem into a sequence of smoothing problems. There are multiple regression equations and we want to select the same strongest genes for all regression equations to constitute a target predictor set or, in the context of a genetic network, the dependency set for the target. The probit regressor is approximated as a linear combination of the genes and a Gibbs sampler is employed to find the strongest genes. Numerical techniques to speed up the computation are discussed. After finding the strongest genes, we predict the target gene based on the strongest genes, with the coefficient of determination being used to measure predictor accuracy. Using malignant melanoma microarray data, we compare two predictor models, the estimated probit regressors themselves and the optimal full-logic predictor based on the selected strongest genes, and we compare these to optimal prediction without feature selection.

Keywords and phrases: gene microarray, multinomial probit regression, Bayesian gene selection, genetic regulatory networks.

1. INTRODUCTION

The advent of high throughput gene expression microarray technology has stimulated the development of mathematical models for genetic regulatory networks, in particular, discrete models such as Bayesian networks [1, 2, 3, 4], Boolean networks [5, 6, 7, 8], probabilistic Boolean networks [9, 10], and the generalization of both deterministic and probabilistic Boolean networks to multilevel quantization [11, 12]. A critical issue for network construction is the identification of network topology from the data. This issue is related to the more general problem of expression prediction in which we want to find small subsets of genes to be used as predictors of target genes [11, 13]. Given some maximum number of predictors to be used, ideally one would like to search over all possible predictor sets to find those that are the best relative to some measure of prediction such as the coefficient of determination [14]; however, such a search is combinatorially prohibitive except for small predictors sets, and even then, may require supercomputing [15]. Consequently, this has lead to an effort to find other, perhaps suboptimal, approaches to finding predictor sets, and the concomitant network topologies. Two such efforts involve minimum description length [16], mutual-information-based clustering [12], and incremental inclusion of predictor variables [17].

The search for good predictor sets is a form of feature reduction, which in the context of expression-based classification involves methods to reduce the set of genes from which

good feature sets can be formed. Owing to the importance of classification and the extremely large number of genes from which to form classifiers from microarray data, several methods have been proposed, including the support vector machine method [18], minimum description length [19], voting [20], and Bayesian variable selection [21, 22].

In this paper, we focus on Bayesian variable selection for prediction using a multinomial regression model (probit regressor) with data augmentation to turn the multinomial problem into a sequence of smoothing problems [23]. In a sense, this work extends the method of [22], except that here the input and output values are ternary instead of analog and binary, respectively. This means that there are multiple regression equations and we want to select the same strongest genes for all regression equations to constitute a target predictor set or, in the context of a genetic regulatory network, the dependency set for the target. The probit regressor is approximated as a linear combination of the genes and a Gibbs sampler is employed to find the strongest genes. Since this method has high computational complexity, we discuss some numerical techniques to speed up the computation. After finding the strongest genes, we predict the target gene based on the strongest genes, with the coefficient of determination being used to measure predictor accuracy. Normally, when trying to identify network topologies and related problems, one uses time series data. In this paper, we aim at the same goal using static data, that is, malignant melanoma microarray data [24]. Using malignant melanoma microarray data, we compare two predictor models: (1) the estimated probit regressors themselves and (2) the optimal full-logic predictor based on the selected strongest genes. As must be the case, full-logic prediction with the strongest genes will outperform the regressor model with the strongest genes; nevertheless, the fundamental issue in this paper is feature reduction and this is accomplished satisfactorily if the optimal full-logic predictor performs well with the selected feature set.

2. MULTINOMIAL PROBIT REGRESSION WITH BAYESIAN GENE SELECTION

2.1. Problem formulation

Assume that there are $n+1$ genes, say, $x_1,\ldots,x_n,x_{n+1}$. Without loss of generality, we assume that the target gene is x_{n+1}, and let w denote this target gene. Then $\mathbf{w} = [w_1,\ldots,w_m]^T$ denotes the normalized expression profiles of the target gene (e.g., for the normalized ternary expression data, $w_j = 1$ indicates that the sample j is up-regulated; $w_j = -1$ indicates that the sample j is down-regulated; and $w_j = 0$ indicates that the sample j is invariant). Denote

$$\mathbf{X} = \begin{bmatrix} \text{Gene 1} & \text{Gene 2} & \cdots & \text{Gene } n \\ x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

as the normalized expression profiles of genes $x_1,\ldots,x_n$. The gene selection problem is to find some genes from $x_1,\ldots,x_n$ that are useful in predicting some target gene w. Here, we consider a more general case of gene prediction, that is, assume that the gene expression profiles are normalized to K levels.

The perceptron has been proved to be an effective model to model the relationship between the target gene and the other genes [25]. Here, we study this problem by using probit regression with Bayesian gene selection. Let $\mathbf{X}_i$ denote the ith row of matrix $\mathbf{X}$ in (1). In the binomial probit regression, that is, when $K = 2$, the relationship between w_i and the gene expression levels $\mathbf{X}_i$ is modeled as a probit regressor [23] which yields

$$P(w_i = 1|\mathbf{X}_i) = \Phi(\mathbf{X}_i\boldsymbol{\beta}), \quad i = 1,\ldots,m, \tag{2}$$

where $\boldsymbol{\beta} = (\beta_1,\beta_2,\ldots,\beta_n)^T$ is the vector of regression parameters and Φ is the standard normal cumulative distribution function. Introduce m independent latent variable $z_1,\ldots,z_m$, where $z_i \sim N(\mathbf{X}_i\boldsymbol{\beta},1)$, that is,

$$z_i = \mathbf{X}_i\boldsymbol{\beta} + e_i, \quad i = 1,\ldots,m, \tag{3}$$

and $e_i \sim N(0,1)$. Define $\boldsymbol{\gamma}$ as the $n\times 1$ indicator vector with the jth element γ_j such that $\gamma_j = 0$ if $\beta_j = 0$ (the variable is not selected) and $\gamma_j = 1$ if $\beta_j \neq 0$ (the variable is selected). The Bayesian variable selection is to estimate $\boldsymbol{\gamma}$ from the posteriori distribution $p(\boldsymbol{\gamma}|\mathbf{z})$. See [11] for details.

However, when $K > 2$, the situation is different from the binomial case because we have to construct $K-1$ regression equations similar to (3). Introduce $K-1$ latent variables $z_1,\ldots,z_{K-1}$ and $K-1$ regression equations such that $z_k = \mathbf{X}\boldsymbol{\beta}_k + e_k$, $k = 1,\ldots,K-1$, where $e_k \sim N(0,1)$. Let z_k take m values $\{z_{k,1},\ldots,z_{k,m}\}$. Using matrix form, it can be further written as

$$\begin{aligned} z_{k,1} &= \mathbf{X}_1\boldsymbol{\beta}_k + e_{k,1}, \\ z_{k,2} &= \mathbf{X}_2\boldsymbol{\beta}_k + e_{k,2}, \\ &\vdots \\ z_{k,m} &= \mathbf{X}_m\boldsymbol{\beta}_k + e_{k,m}, \end{aligned} \tag{4}$$

where $k = 1,\ldots,K-1$. Denote $\mathbf{z}_k \triangleq [z_{k,1},\ldots,z_{k,m}]^T$ and $\mathbf{e}_k \triangleq [e_{k,1},\ldots,e_{k,m}]^T$. Then (4) can be rewritten as

$$\mathbf{z}_k = \mathbf{X}\boldsymbol{\beta}_k + \mathbf{e}_k, \quad k = 1,\ldots,K-1. \tag{5}$$

This model is called the multinomial probit model. For background on multinomial probit models, see [26]. Note that we do not have the observations of $\{\mathbf{z}_k\}_{k=1}^{K-1}$, which makes it difficult to estimate the parameters in (5).

Here, we discuss how to select the same strongest genes for the different regression equations. The model is a little different from (5), that is, the selected genes do not change with the different regression equations. Note that the

(i) Draw $\boldsymbol{\gamma}$ from $p(\boldsymbol{\gamma}|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1})$. We usually sample each γ_i independently from

$$\begin{aligned} &p(\gamma_i|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1},\gamma_{j\neq i}) \\ &\quad\propto p(\mathbf{z}_1,\ldots,\mathbf{z}_{K-1}|\boldsymbol{\gamma})\,p(\gamma_i) \\ &\quad\propto (1+c)^{-(K-1)n_\gamma/2}\exp\left\{-\frac{1}{2}\sum_{k=1}^{K-1}S(\boldsymbol{\gamma},\mathbf{z}_k)\right\}\pi_i^{\gamma_i}(1-\pi_i)^{1-\gamma_i}, \end{aligned} \tag{10}$$

$n_\gamma=\sum_{j=1}^{n}\gamma_j$, $c=10$, and $\pi_i=P(\gamma_i=1)$ are prior probabilities to select the jth gene. It is set as $\pi_i=8/n$ according to the very small sample size. If π_i takes a larger value, we find oftentimes that $(\mathbf{X}_\gamma{}^T\mathbf{X}_\gamma)^{-1}$ does not exist.

(ii) Draw $\boldsymbol{\beta}_k$ from

$$p(\boldsymbol{\beta}_k|\boldsymbol{\gamma},\mathbf{z}_k)\propto\mathcal{N}(\mathbf{V}_\gamma\mathbf{X}_\gamma{}^T\mathbf{z}_k,\mathbf{V}_\gamma), \tag{11}$$

where $\mathbf{V}_\gamma=(c/(1+c))(\mathbf{X}_\gamma{}^T\mathbf{X}_\gamma)^{-1}$.

(iii) Draw $\mathbf{z}_k=[z_{k,1},\ldots,z_{k,m}]^T$, $k=1,\ldots,K$, from a truncated normal distribution as follows [27].

For $i=1,2,\ldots,m$

If $w_i=k$, then draw $z_{k,i}$ according to $z_{k,i}\sim N(\mathbf{X}_\gamma\boldsymbol{\beta}_k,1)$ truncated left by $\max_{j\neq k}z_{j,i}$, that is,

$$z_{k,i}\sim\mathcal{N}(\mathbf{X}_\gamma\boldsymbol{\beta}_k,1)1_{\{z_{k,i}>\max_{j\neq k}z_{j,i}\}}. \tag{12}$$

Else $w_i=j$ and $j\neq k$, then draw $z_{j,i}$ according to $z_{j,i}\sim N(\mathbf{X}_\gamma\boldsymbol{\beta}_j,1)$ truncated right by the newly generated $z_{k,i}$, that is,

$$z_{j,i}\sim\mathcal{N}(\mathbf{X}_\gamma\boldsymbol{\beta}_j,1)1_{\{z_{j,i}\leq z_{k,i}\}}. \tag{13}$$

Endfor.

Here, we set $z_{K,i}\sim N(0,1)$ when $w_i=K$, that is, we introduce a new equation $z_{K,i}=\mathbf{X}_\gamma\boldsymbol{\beta}_K+e_{K,i}$, $i=1,\ldots,m$, with $\boldsymbol{\beta}_K$ being a zero vector and $e_{K,i}\sim N(0,1)$.

Algorithm 1

parameter $\boldsymbol{\beta}$ is still dependent on k and $\boldsymbol{\gamma}$, denoted by $\boldsymbol{\beta}_{k,\gamma}$. Then (5) is rewritten as

$$\mathbf{z}_k=\mathbf{X}_\gamma\boldsymbol{\beta}_{k,\gamma}+\mathbf{e}_k,\quad k=1,\ldots,K-1, \tag{6}$$

where $\mathbf{X}_\gamma$ means the column of $\mathbf{X}$ corresponding to those elements of $\boldsymbol{\gamma}$ that are equal to 1, and the same applies to $\boldsymbol{\beta}_{k,\gamma}$. Now, the problem is how to estimate $\boldsymbol{\gamma}$ and the corresponding $\boldsymbol{\beta}_{k,\gamma}$ and $\mathbf{z}_k$ for each equation in (6).

2.2. Bayesian variable selection

A Gibbs sampler is employed to estimate all the parameters. Given $\boldsymbol{\gamma}$ for equation k, the prior distribution of $\boldsymbol{\beta}_\gamma$ is $\boldsymbol{\beta}_\gamma\sim N(0,c(\mathbf{X}_\gamma^T\mathbf{X}_\gamma)^{-1})$ [22], where c is a constant (we set $c=10$ in this study). The detailed derivation of the posterior distributions of the parameters are given in [22]. Here, we summarize the procedure for Bayesian variable selection. Denote

$$S(\boldsymbol{\gamma},\mathbf{z}_k)=\mathbf{z}_k^T\mathbf{z}_k-\frac{c}{c+1}\mathbf{z}_k^T\mathbf{X}_\gamma(\mathbf{X}_\gamma{}^T\mathbf{X}_\gamma)^{-1}\mathbf{X}_\gamma{}^T\mathbf{z}_k, \tag{7}$$

where $k=1,\ldots,K-1$. Then the Gibbs sampling algorithm for estimating $\{\boldsymbol{\gamma},\boldsymbol{\beta}_k,\mathbf{z}_k\}$ is as follows. By straightforward computing, the posteriori distribution $p(\boldsymbol{\gamma}|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1})$ is approximated by

$$\begin{aligned} &p(\boldsymbol{\gamma}|\mathbf{z}_1,\ldots,\mathbf{z}_{K-1}) \\ &\quad\propto p(\mathbf{z}_1,\ldots,\mathbf{z}_{K-1}|\boldsymbol{\gamma})\,p(\boldsymbol{\gamma}) \\ &\quad\propto (1+c)^{-(K-1)n_\gamma/2} \\ &\qquad\times\exp\left\{-\frac{1}{2}\sum_{k=1}^{K-1}S(\boldsymbol{\gamma},\mathbf{z}_k)\right\}\prod_{i=1}^{n}\pi_i^{\gamma_i}(1-\pi_i)^{1-\gamma_i}, \end{aligned} \tag{8}$$

and the posterior distribution $p(\boldsymbol{\beta}_{k,\gamma}|\mathbf{z}_k)$ is given by

$$\boldsymbol{\beta}_{k,\gamma}|\mathbf{z}_k,\mathbf{X}_\gamma\sim N(\mathbf{V}_\gamma\mathbf{X}_\gamma{}^T\mathbf{z}_k,\mathbf{V}_\gamma). \tag{9}$$

The Gibbs sampling algorithm for estimating $\boldsymbol{\gamma}$, $\{\boldsymbol{\beta}_{k,\gamma}\}$, and $\{\mathbf{z}_k\}$ is illustrated in Algorithm 1.

In this study, 12000 Gibbs iterations are implemented with the first 2000 as burn-in period. Then we obtain the Monte Carlo samples as $\boldsymbol{\gamma}^{(t)},\boldsymbol{\beta}_k^{(t)},\mathbf{z}_k^{(t)}$, $t=2001,\ldots,T$, where $T=10000$. Finally, we count the number of times that each gene appears in $\boldsymbol{\gamma}^{(t)}$, $t=2001,2002,\ldots,T$. The genes with the highest appearance frequencies play the strongest role in predicting the target gene. We will discuss some implementation issues of Algorithm 1 in Section 3.

2.3. Bayesian estimation using the strongest genes

Now, assume that the genes corresponding to nonzeros of $\boldsymbol{\gamma}$ are the strongest genes obtained by Algorithm 1. For fixed $\boldsymbol{\gamma}$, we again use a Gibbs sampler to estimate the probit regression coefficients $\boldsymbol{\beta}_k$ as follows: first, draw $\boldsymbol{\beta}_{k,\boldsymbol{\gamma}}$ according to (11), then draw $\mathbf{z}_k$ and iterate the two steps. In this study, 1500 iterations are implemented with the first 500 as the burn-in period. Thus, we obtain the Monte Carlo samples $\boldsymbol{\beta}_{k,\boldsymbol{\gamma}}^{(t)}, \mathbf{z}_k^{(t)}$, $t = 501, \ldots, \tilde{T}$. The probability of a given sample $\mathbf{x}$ under each class is given by

$$P(w = k|\mathbf{x}) = \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \prod_{j=1,\, j\neq k}^{K} \Phi\left(\mathbf{x}_{\boldsymbol{\gamma}} \boldsymbol{\beta}_{k,\boldsymbol{\gamma}}^{(t)} - \mathbf{x}_{\boldsymbol{\gamma}} \boldsymbol{\beta}_{j,\boldsymbol{\gamma}}^{(t)}\right), \quad k = 1, \ldots, K-1, \tag{14}$$

$$P(w = K|\mathbf{x}) = 1 - \sum_{k=1}^{K-1} P(w = k|\mathbf{x}), \tag{15}$$

where $\boldsymbol{\beta}_{K,\boldsymbol{\gamma}}^{(t)}$ is a zero vector; and the estimation of this sample is given by

$$\hat{w} \triangleq d(w) = \arg \max_{1 \le k \le K} P(w = k|\mathbf{x}). \tag{16}$$

Note that (15) may be computed using another formulation, which is replaced by [28, (13)].

In order to measure the fitting accuracy of such a predictor, we next define the coefficient of determination (COD) for this probit predictor. In fact, the above $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ (including all parameters $\boldsymbol{\beta}_{k,\boldsymbol{\gamma}}$) are dependent on the target gene w. Firstly, a probabilistic error measure $\epsilon(w, \mathbf{x}_{\boldsymbol{\gamma}}, \boldsymbol{\beta})$ associated with the predictors $\boldsymbol{\gamma}, \boldsymbol{\beta}$ is defined as

$$\epsilon(w, \mathbf{x}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}) \triangleq \mathbb{E}\left[\, |\, d(w) - w\, |^2\right], \tag{17}$$

where $\mathbb{E}$ denotes the expectation. Similar to the definition in [14], the COD for w relative to the conditioning sets $\boldsymbol{\gamma}, \boldsymbol{\beta}$ is defined by

$$\theta = \frac{\epsilon - \epsilon(w, \mathbf{x}_{\boldsymbol{\gamma}}, \boldsymbol{\beta})}{\epsilon}, \tag{18}$$

where ϵ is the error of the best (constant) estimate of w in the absence of any conditional variables. In the case of minimum mean square error estimation, ϵ is defined as

$$\epsilon = \mathbb{E}[\, |\, w - g(\mathbb{E}(w))\, |^2], \tag{19}$$

where g is a $\{-1, 0, 1\}$-valued threshold function [$g(z) = 0$ if $-0.5 < z < 0.5$, $g(z) = 1$ if $z \ge 0.5$, and $g(z) = -1$ if $z \le -0.5$] for ternary data.

3. FAST IMPLEMENTATION ISSUES

The computational complexity of the Bayesian gene selection algorithm in (Algorithm 1) is very high. For example, if there are 1000 gene variables, then for each iteration, we have to compute the matrix inverse $(\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1}$ 1000 times because we need to compute (10) for each gene. Hence, some fast algorithms must be developed to deal with the problem.

3.1. Preselection method

When there is a very large number of genes, we employ a preselection method. In pattern recognition, the following criterion is often adopted: the smaller is the sum of squares within groups and the bigger is the sum of squares between groups, the better is the classification accuracy. Therefore, we can define a score using the above two statistics to preselect genes, that is, the ratio of the between-group to within-group sum of squares. It is not necessary to adopt this procedure if the number of genes is small.

3.2. Computation of $p(\gamma_j|\mathbf{z}_k, \gamma_{i\neq j})$ in (10)

Because γ_j only takes 0 or 1, we can take a close look at $p(\gamma_j = 1|\mathbf{z}_k,\ i \neq j)$ and $p(\gamma_j = 0|\mathbf{z}_k,\ i \neq j)$. Let

$$\begin{aligned} \boldsymbol{\gamma}^1 &= (\gamma_1, \ldots, \gamma_{j-1},\ \gamma_j = 1,\ \gamma_{j+1}, \ldots, \gamma_n), \\ \boldsymbol{\gamma}^0 &= (\gamma_1, \ldots, \gamma_{j-1},\ \gamma_j = 0,\ \gamma_{j+1}, \ldots, \gamma_n). \end{aligned} \tag{20}$$

After a straightforward computation of (10), we have

$$p(\gamma_j = 1|\mathbf{z}_k,\ \gamma_{i\neq j}) \propto \frac{1}{1+h}, \tag{21}$$

with

$$h = \frac{1-\pi_j}{\pi_j} \exp\left\{\frac{S(\boldsymbol{\gamma}^1, \mathbf{z}_k) - S(\boldsymbol{\gamma}^0, \mathbf{z}_k)}{2}\right\} \sqrt{1+c}. \tag{22}$$

If $\boldsymbol{\gamma} = \boldsymbol{\gamma}^0$ before γ_j is generated, this means that we have obtained $S(\boldsymbol{\gamma}^0, \mathbf{z}_k)$, then we only need to compute $S(\boldsymbol{\gamma}^1, \mathbf{z}_k)$ and vice versa.

3.3. Fast computation of $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ in (7)

From the above discussion, it is a key step to compute $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ fast when a gene variable is added or removed from $\boldsymbol{\gamma}$. Denote

$$E(\boldsymbol{\gamma}, \mathbf{z}_k) = \mathbf{z}_k^T \mathbf{z}_k - \mathbf{z}_k^T \mathbf{X}_{\boldsymbol{\gamma}} (\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1} \mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{z}_k, \tag{23}$$

where $k = 1, \ldots, K-1$. Then (23) can be computed using the fast QR-decomposition, QR-delete, and QR-insert algorithms when a variable is added or removed [29, Chapter 10.1.1b]. Now, we want to estimate $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ in (7). Comparing (23) and (7), one can obtain the following equation:

$$\mathbf{z}_k^T \mathbf{X}_{\boldsymbol{\gamma}} (\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1} \mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{z}_k = (1+c)[S(\boldsymbol{\gamma}, \mathbf{z}_k) - E(\boldsymbol{\gamma}, \mathbf{z}_k)]. \tag{24}$$

Substituting (24) into (7), after a straightforward computation, $S(\boldsymbol{\gamma}, \mathbf{z}_k)$ is given by

$$S(\boldsymbol{\gamma}, \mathbf{z}_k) = \frac{\mathbf{z}_k^T \mathbf{z}_k + cE(\boldsymbol{\gamma}, \mathbf{z}_k)}{1+c}, \quad k = 1, \ldots, K-1. \tag{25}$$

(i) Preselect genes.

(ii) Initialization: Randomly set initial parameters $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{\beta}_k^{(0)}$, $\mathbf{z}_k^{(0)}$.

(iii) For $t = 1, 2, \ldots, 12000$
  Draw $\boldsymbol{\gamma}^{(t)}$. For $j = 1, \ldots, n$
    Compute $S(\boldsymbol{\gamma}^{(t)}, \mathbf{z}_k)$ using QR-delete or QR-insert.
    Compute $p(\gamma_j = 1 | \mathbf{z}_k, \gamma_{i \neq j}^{(t)})$ according to (21).
    Draw $\gamma_j^{(t)}$ from $p(\gamma_j = 1 | \mathbf{z}_k^{(t-1)}, \gamma_{i \neq j}^{(t)})$.
  Draw $\boldsymbol{\beta}_k^{(t)}$ according to (11);
  Draw $\mathbf{z}_k^{(t)}$ according to (12) and (13).

(iv) Endfor.

(v) Count the frequency of each gene appeared in $\boldsymbol{\gamma}^{(t)}$, $t = 2001, \ldots, 12000$.

Algorithm 2

Thus, after computing $E(\boldsymbol{\gamma}, \mathbf{z}_k)$ using QR-decomposition, QR-delete, and QR-insert algorithms, we then obtain $S(\boldsymbol{\gamma}, \mathbf{z}_k)$. Here, we only need to compute the matrix inverse one time each iteration, but in the original algorithm, we have to compute the matrix inverse for n time each iteration. The computation complexity will be much smaller than that of the original algorithm [22] due to our processing techniques. To that end, we summarize our fast Bayesian gene selection algorithm as in Algorithm 2.

Notice that if it happens that the number of selected genes is more than the total number of samples, we need to remove this case because $(\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})^{-1}$ does not exist. Another concern is that if it happens that $(\mathbf{X}_{\boldsymbol{\gamma}}{}^T \mathbf{X}_{\boldsymbol{\gamma}})$ is singular due to some rows or columns being a constant, then we need to add a very small random number to each element in $\mathbf{X}_{\boldsymbol{\gamma}}$.

4. EXPERIMENTAL RESULTS

In the first step in constructing a gene regulatory network, the complexity of the expression data is reduced by thresholding changes in transcript level into ternary expression data: -1 (down-regulated), $+1$ (up-regulated), or 0 (invariant). When using multiple microarrays, the absolute signal intensities vary extensively due to both the process of preparing and printing the EST elements [30] and the process of preparing and labeling the cDNA representations of the RNA pools. This problem is solved via internal standardization. We then build gene regulatory networks using the proposed approaches.

4.1. Malignant melanoma microarray data

The gene expression profiles used in this study result from a study of 31 malignant melanoma samples [24]. For the study, total messenger RNA was isolated directly from melanoma biopsies. Fluorescent cDNA from the message was prepared and hybridized to a microarray containing probes for 8 150 cDNAs (representing 6 971 unique genes). A set of 587 genes has been subjected to an analysis of their ability to cross predict each other's state in a multivariate setting [11, 13, 25]. From these, we have selected 26 differential genes using the following t-test:

$$t(j) = \frac{\bar{x}_{1,j} - \bar{x}_{2,j}}{s_0(j)\sqrt{1/m_1 + 1/m_2}}, \quad j = 1, \ldots, p, \tag{26}$$

with

$$s_0(j) \triangleq \sqrt{\frac{(m_1 - 1)s_1(j)^2 + (m_2 - 1)s_2(j)^2}{m_1 + m_2}}, \tag{27}$$

where p is the number of genes, $\{\bar{x}_{k,j}\}_{k=1}^2$ denotes the average expression level of gene j across the samples belonging to class k, m_1 and m_2 are the numbers of the two classes, and $\{s_k(j)^2\}_{k=1}^2$ are the variances of gene j across the samples belonging to class k. Genes with $t(j) \geq 0.05$ are listed in Table 1.

COD values for all the 26 targets have been computed using the strongest genes found via the Bayesian selection. CODs have been computed using leave-one-out cross validation. The strongest genes for each target are listed in the second column of Table 2 and the third column lists the CODs using the top 2, 3, and 4 genes for each target and using the probit regression to form the predictors. Several points should be noted. First, while the theoretical (distributional) COD values increase as the number of predictors increases, this is not necessarily the case for experimental data, especially when small samples are involved (on account of overfitting and high variance of cross-validation error estimation). Second, pirin (no. 2) is a strong predictor gene in many cases, and this agrees with the comment in the original paper that pirin has a very high discriminative weight [24]. Third, even with feature selection and a suboptimal predictor function, for the most part, the CODs are fairly high.

Having made the last point, we note that our salient interest is gene selection. Hence, having found strong genes via Bayesian variable selection, we are not compelled to use the probit regression model to form the predictors; rather, we can choose the optimal predictor using the strong genes among all possible (full-logic) predictor functions. We can

Table 1: The 26 differential genes.

Gene no.	Index no.	Gene description
1	3	Tumor protein D52
2	7	Pirin
3	14	V-myc avian myelocytomatosis viral oncogene homolog
4	42	Endothelin receptor type B
5	60	ESTS
6	79	Alpha-2-macroglobulin
7	117	V-myc avian myelocytomatosis viral oncogene homolog
8	126	ESTs
9	175	Myotubularin related protein 4
10	210	NGFI-A binding protein 2 (ERG1 binding protein 2)
11	216	IQ motif containing GTPase activating protein 1
12	220	Annexin A2
13	228	ESTs
14	245	Homo sapiens mRNA; cDNA DKFZp434L057 (from clone DKFZp434L057)
15	282	Endothelin receptor type B
16	292	ESTs
17	323	ESTs
18	360	Glycoprotein M6B
19	372	"Nuclear receptor subfamily 4, group A, member 3"
20	374	Thrombospondin 2
21	387	"ESTs, weakly similar to HP1-BP74 protein [M.musculus]"
22	404	"Phosphofructokinase, liver"
23	506	Placental transmembrane protein
24	556	Human insulin-like growth factor binding protein 5 (IGFBP5) mRNA
25	573	"Platelet-derived growth factor receptor, alpha polypeptide"
26	576	ESTs

Table 2: Strongest genes to predict each gene and the corresponding COD values for 2, 3, and 4 predictor genes.

Target gene no.	Strongest genes (no.)				COD		
	1	2	3	4	2	3	4
1	19	23	22	17	0.6452	0.6129	0.7097
2	25	1	19	11	0.3871	0.6774	0.8065
3	7	23	2	5	0.7097	0.7742	0.7742
4	15	2	13	17	0.7419	0.7742	0.8710
5	14	2	13	10	0.5484	0.5161	0.4194
6	10	2	19	24	0.6129	0.7097	0.8387
7	3	2	17	1	0.7419	0.8387	0.8387
8	20	2	21	14	0.5161	0.5484	0.5484
9	2	13	17	15	0.6774	0.7097	0.7742
10	6	20	2	4	0.6129	0.6452	0.6774
11	13	25	2	1	0.8710	0.8710	0.7742
12	2	13	11	14	0.6452	0.6452	0.7419
13	2	15	11	18	0.8387	1.0000	1.0000
14	2	25	21	15	0.6774	0.7742	0.6774
15	2	4	13	14	0.8065	0.7419	0.9677
16	4	25	2	7	0.6452	0.7097	0.6452
17	11	18	2	8	0.8387	0.8065	0.8387
18	2	17	13	23	0.8387	0.7742	0.8710
19	1	22	2	9	0.7419	0.6774	0.7419
20	22	5	10	24	0.3548	0.3548	0.7419
21	25	2	14	20	0.7742	0.7742	0.7742
22	2	9	6	23	0.6774	0.7097	0.7742
23	24	2	1	5	0.5161	0.5484	0.6774
24	2	20	3	7	0.5806	0.6129	0.6452
25	11	2	14	13	0.7742	0.6774	0.8065
26	17	13	2	23	0.7742	0.7742	0.8387

Table 3: Three-predictor COD values using full-logic predictor, full search, and Bayesian-selected genes. There are 2300 three-predictor sets for each target gene.

Target gene no.	Probit position	logic COD (best)	logic COD (probit)
1	32	0.8065	0.7419
2	59	0.8387	0.7419
3	36	0.9355	0.9032
4	15	0.9677	0.9032
5	52	0.7742	0.6774
6	1	0.9677	0.9677
7	30	0.9355	0.9032
8	91	0.8387	0.7419
9	141	0.8710	0.7742
10	25	0.9677	0.9032
11	49	0.9677	0.8710
12	173	0.8387	0.7419
13	1	1.0000	1.0000
14	212	0.8387	0.7419
15	102	0.9677	0.9355
16	46	0.8710	0.7742
17	12	0.9677	0.9355
18	289	0.9355	0.8710
19	196	0.9677	0.8387
20	21	0.8710	0.8387
21	14	0.8387	0.8065
22	16	0.9355	0.9032
23	48	0.9032	0.8065
24	29	0.8065	0.7097
25	69	0.8710	0.7742
26	49	0.9355	0.9032

also compare the COD for this approach with the fully optimal COD derived from considering all possible predictor sets from among the full-gene set and all possible predictor functions. The results of this analysis for three predictor variables are shown in Table 3. For each target, the second column gives the rank of the COD resulting from the probit predictors in the list of all the 2300 CODs found from all possible subsets of three predictors using the best full-logic predictor. The selected gene sets rank very high except in a couple of cases. The third and fourth columns give the CODs for the best full-logic predictor with a full search of the gene subsets and the best full-logic predictor using the strongest three genes found by Bayesian gene selection. As must be the case, the values in the third column must exceed the values in the fourth, but in general, this does not happen much, even when the probit-selected predictor set does not rank near the top. The differences are likely due to multivariate interaction between the predictors not recognized by the sequential selection of strongest genes [17]. Table 4 shows analogous results for four predictors. For it, we note that there are 12 650 predictor sets for each target. Similar comments apply to the genes in Table 4.

It is interesting to compare the fourth column in Table 4 with the third in Table 3. For large gene sets (say, 600 to 1000 genes), a full search over all the three-variable predictor sets is feasible with a supercomputer running for weeks [15]. But a full search is not feasible for a full search over all four-variable predictor sets. Optimal four-connectivity may not be possible in network design. Hence, the small loss in COD between the full-search column in Table 3 and the probit-selection column in Table 4 demonstrates the potential of the Bayesian feature selection. Indeed, there are a number of cases in which the four-variable probit-selected genes outperform the corresponding three-variable full-search genes. Just to get an idea of the vast difference between the methods, the Gibbs sampler would need approximately 12000×1000 iterations, whereas the fully optimal full-search predictor would need to consider 2^{1000} predictor sets. Even for four-variable predictor sets, the full search needs C_4^{1000} iterations, which is vastly larger than the Gibbs sampling search.

Table 4: Four-Predictor COD values using full-logic predictor, full search, and Bayesian-selected genes. There are 12650 four-predictor sets for each target gene.

Target gene no.	Probit position	Logic COD (best)	Logic COD (probit)
1	48	0.8710	0.7742
2	70	0.8710	0.8065
3	14	0.9677	0.9355
4	283	1.0000	0.9355
5	48	0.8387	0.7419
6	1	0.9677	0.9677
7	82	0.9677	0.9032
8	101	0.8710	0.7742
9	60	0.9032	0.8387
10	569	0.9677	0.8710
11	82	0.9677	0.9032
12	510	0.9355	0.8065
13	1	1.0000	1.0000
14	131	0.8710	0.8065
15	1	1.0000	1.0000
16	60	0.8710	0.8065
17	65	0.9355	0.8710
18	364	0.9677	0.8710
19	170	0.8065	0.7419
20	52	0.9355	0.8387
21	193	0.9355	0.9032
22	163	0.9677	0.9032
23	240	0.9677	0.8710
24	91	0.8065	0.7419
25	58	0.9032	0.8387
26	79	0.9677	0.9355

5. CONCLUSION

We have studied the problem of multilevel gene prediction and genetic network construction from gene expression data based on multinomial probit regression with Bayesian gene selection, which selects genes closely related to a particular target gene. Some fast implementation issues for this Bayesian gene selection method have been discussed, in particular, computing estimation errors recursively using QR decomposition. Experimental results using malignant melanoma data show that the Bayesian gene selection yields predictor sets with coefficients of determination that are competitive with those obtained via a full search over all possible predictor sets.

ACKNOWLEDGMENTS

This research was supported by the National Human Genome Research Institute and the Translational Genomics Research Institute. X. Wang was supported in part by the US National Science Foundation under Grant DMS-0225692.

REFERENCES

[1] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Baysian networks to analyze expression data," *Computational Biology*, vol. 7, no. 3/4, pp. 601–620, 2000.

[2] E. J. Moler, D. C. Radisky, and I. S. Mian, "Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in S. cerevisiae," *Physiological Genomics*, vol. 4, no. 2, pp. 127–135, 2000.

[3] K. Murphy and S. Mian, "Modelling gene expression data using dynamic Bayesian networks," Tech. Rep., University of California, Berkeley, Calif, USA, 1999, http://citeseer.nj.nec.com/murphy99modelling.html.

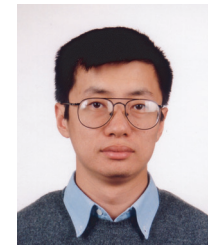
[4] D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed expression profiles," *Bioinformatics*, vol. 17, suppl. 1, pp. S215–S224, 2001.

[5] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under Boolean network model," in *Proc. Pacific Symposium on Biocomputing*, vol. 4, pp. 17–28, Maui, Hawaii, USA, January 1999.

[6] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[7] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorgenesis and drug discovery," *Molecular Medicine*, vol. 77, no. 6, pp. 469–480, 1999.
[8] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, NY, USA, 1993.
[9] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.
[10] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.
[11] S. Kim, H. Li, E. R. Dougherty, et al., "Can Markov chain models mimic biological regulation?," *Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.
[12] X. Zhou, X. Wang, and E. R. Dougherty, "Construction of genomic networks using mutual-information clustering and reversible-jump Markov-Chain-Monte-Carlo predictor design," *Signal Processing*, vol. 83, no. 4, pp. 745–761, 2003.
[13] S. Kim, E. R. Dougherty, Y. Chen, et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.
[14] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.
[15] E. B. Suh, E. R. Dougherty, S. Kim, D. E. Russ, and R. L. Martino, "Parallel computing methods for analyzing gene expression relationships," in *Proc. SPIE Microarrays: Optical Technologies and Informatics*, San Jose, Calif, USA, January 2001.
[16] I. Tabus and J. Astola, "On the use of MDL principle in gene expression prediction," *Applied Signal Processing*, vol. 2001, no. 4, pp. 297–303, 2001.
[17] R. F. Hashimoto, E. R. Dougherty, M. Brun, Z.-Z. Zhou, M. L. Bittner, and J. M. Trent, "Efficient selection of feature sets possessing high coefficients of determination based on incremental determinations," *Signal Processing*, vol. 83, no. 4, pp. 695–712, 2003.
[18] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.
[19] R. Jornsten and B. Yu, "Simultaneous gene clustering and subset selection for sample classification via MDL," *Bioinformatics*, vol. 19, no. 9, pp. 1100–1109, 2003.
[20] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[21] H. Chipman, E. I. George, and R. McCulloch, "The practical implementation of Bayesian model selection," in *Model Selection*, vol. 38, pp. 65–134, Institute of Mathematical Statistics, Hayward, Calif, USA, 2001.
[22] K. E. Lee, N. Sha, E. R. Dougherty, M. Vannucci, and B. K. Mallick, "Gene selection: a Bayesian variable selection approach," *Bioinformatics*, vol. 19, no. 1, pp. 90–97, 2003.
[23] J. Albert and S. Chib, "Bayesian analysis of binary and polychotomous response data," *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 669–679, 1993.
[24] M. Bittner, P. Meltzer, Y. Chen, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.
[25] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *Biomedical Optics*, vol. 5, no. 4, pp. 411–424, 2000.
[26] K. Imai and D. A. van Dyk, "A Bayesian analysis of the multinomial probit model using marginal data augmentation," http://www.princeton.edu/~kimai/research/mnp.html.
[27] C. P. Robert, "Simulation of truncated normal variables," *Statistics and Computing*, vol. 5, pp. 121–125, 1995.
[28] P. Yau, R. Kohn, and S. Wood, "Bayesian variable selection and model averaging in high-dimensional multinomial nonparametric regression," *Computational and Graphical Statistics*, vol. 12, no. 1, pp. 23–54, 2003.
[29] G. A. F. Seber, *Multivariate Observations*, John Wiley & Sons, NY, USA, 1984.
[30] Y. Chen, E. R. Dougherty, and M. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, no. 4, pp. 364–374, 1997.

Xiaobo Zhou received the B.S. degree in mathematics from Lanzhou University, Lanzhou, China, in 1988, the M.S. and the Ph.D. degrees in mathematics from Peking University, Beijing, China, in 1995 and 1998, respectively. From 1988 to 1992, he was a Lecturer at the Training Center in the 18th Building Company, Chongqing, China. From 1992 to 1998, he was a Research Assistant and Teaching Assistant in the Department of Mathematics at Peking University, Beijing, China. From 1998 to 1999, he was a postdoctoral fellow in the Department of Automation at Tsinghua University, Beijing, China. From January 1999 to February 2000, he was a Senior Technical Manager of the 3G Wireless Communication Department at Huawei Technologies Co., Ltd., Beijing. From February 2000 to December 2000, he was a postdoctoral fellow in the Department of Computer Science at the University of Missouri-Columbia, Columbia, Mo. From January 2001 to September 2003, he was a postdoctoral fellow in the Department of Electrical Engineering at Texas A&M University, College Station, Tex. Since October 2003, he has been a postdoctoral fellow in the Harvard Center for Neurodegeneration and Repair in Harvard University Medical School and Radiology Department in Brigham and Women's Hospital. His current research interests include bioinformatics in genetics, protein structure informatics, imaging genetics, and gene transcriptional regulatory networks.



Xiaodong Wang received the B.S. degree in electrical engineering and applied mathematics (with the highest honor) from Shanghai Jiao Tong University, Shanghai, China, in 1992; the M.S. degree in electrical and computer engineering from Purdue University in 1995; and the Ph.D. degree in electrical engineering from Princeton University in 1998. From July 1998 to December 2001, he was an Assistant Professor in the Department of Electrical Engineering, Texas A&M University. In January 2002, he joined the Department of Electrical Engineering, Columbia University, as an Assistant Professor. Dr. Wang's research interests fall in the general areas of computing, signal processing, and communications. He has worked in the areas of digital communications, digital signal processing, parallel and distributed

computing, nanoelectronics, and bioinformatics, and has published extensively in these areas. His current research interests include wireless communications, Monte Carlo based statistical signal processing, and genomic signal processing. Dr. Wang received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an Associate Editor for the IEEE Transactions on Communications, the IEEE Transactions on Wireless Communications, the IEEE Transactions on Signal Processing, and the IEEE Transactions on Information Theory.

Edward R. Dougherty is a Professor in the Department of Electrical Engineering at Texas A&M University in College Station. He holds an M.S. degree in computer science from Stevens Institute of Technology in 1986 and a Ph.D. degree in mathematics from Rutgers University in 1974. He is the author of eleven books and the editor of other four books. He has published more than one hundred journal papers, is an SPIE Fellow, and has served as an Editor of the Journal of Electronic Imaging for six years. He is currently Chair of the SIAM Activity Group on Imaging Science. Prof. Dougherty has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His current research focuses on genomic signal processing, with the central goal being to model genomic regulatory mechanisms. He is Head of the Genomic Signal Processing Laboratory at Texas A&M University.

EURASIP Journal on Applied Signal Processing 2004:1, 125–131

Reduction Mappings between Probabilistic Boolean Networks

Ivan Ivanov
Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA
Email: ivanov@ee.tamu.edu

Edward R. Dougherty
Department of Electrical Engineering, Texas A&M University, 3128 TAMU College Station, TX 77843-3128, USA
Email: e-dougherty@tamu.edu

Received 11 April 2003; Revised 28 August 2003

Probabilistic Boolean networks (PBNs) comprise a model describing a directed graph with rule-based dependences between its nodes. The rules are selected, based on a given probability distribution which provides a flexibility when dealing with the uncertainty which is typical for genetic regulatory networks. Given the computational complexity of the model, the characterization of mappings reducing the size of a given PBN becomes a critical issue. Mappings between PBNs are important also from a theoretical point of view. They provide means for developing a better understanding about the dynamics of PBNs. This paper considers two kinds of mappings reduction and projection and their effect on the original probability structure of a given PBN.

Keywords and phrases: Boolean network, genetic network, graphical models, projection, reduction.

1. INTRODUCTION

Given a set of genes, the evolution of their expressions constitutes a dynamical system over time. Owing to the complexity of gene interaction and the paucity of data, homogeneous transitions are customarily assumed. Many different gene-regulatory-network models have been proposed. Among deterministic dynamical systems, perhaps, the most attention has been given to the Boolean network model [1, 2, 3]. In this model, gene expression is quantized to only two levels: ON and OFF. The expression level (state) of a gene is functionally related, via a logical rule, to the expression states of some other genes. The Boolean network model has yielded insights into the overall behavior of large genetic networks [4, 5, 6, 7], thereby facilitating the study of large data sets in a global fashion. Here, we are concerned with a stochastic extension of the Boolean model that results in probabilistic Boolean networks [8, 9]. For these, similarities exist with Bayesian networks [10, 11, 12, 13] and, more generally, with models including stochastic components on the molecular level [14, 15, 16].

The dynamical behavior of such networks can be used to model many biologically meaningful phenomena—for instance, cellular state dynamics, possessing switch-like behavior, stability, and hysteresis [17]. Besides the conceptual framework offered by such models, there are practical uses, such as the identification of suitable drug targets in cancer therapy or inferring the structure of the genetic models from experimental data, for example, from the gene expression profiles [17]. To that end, a significant effort has gone into identifying the structure of gene regulatory networks from expression data [8, 18, 19, 20, 21, 22, 23].

Probabilistic Boolean networks (PBNs) [8, 9] constitute a probabilistic generalization of Boolean networks and offer a more powerful and flexible modeling framework. They share the appealing rule-based properties of the Boolean networks, are robust to uncertainty both in the data and model selection, and can be studied in the probabilistic context of Markov chains (see also [23]). PBNs enable the systematic study of global network dynamics and permit quantification of the relative influence and sensitivity of genes in their interactions with other genes. While the Boolean assumption is useful for a simple up- or down-regulated model and also useful for reducing the complexity of the network, the basic model extends directly to a finite-state-space model, and inference has been studied in that context in [22].

A principle reason for studying regulatory models is to develop intervention strategies to help in guiding the time evolution of the network towards more desirable states. Three distinct approaches to the intervention problem have been considered in the context of PBNs by exploiting their Markovian nature. First, one can toggle the expression status

of a particular gene from ON to OFF or vice versa to facilitate the transition to some other desirable state or set of states. Specifically, using the concept of the mean first passage time, it has been demonstrated how the particular gene, whose transcription status is to be momentarily altered to initiate the state transition, can be chosen to "minimize" (in a probabilistic sense) the time required to achieve the desired state transitions [24]. A second approach has aimed at changing the steady-state (long-run) behavior of the network by minimally altering its rule-based structure [25]. A third approach has focused on applying ideas from control theory to develop an intervention strategy in the general context of Markovian genetic regulatory networks whose state transition probabilities depend on an external (control) variable [26].

An obstacle in applying PBNs is the computational complexity of the model. Owing to the large number of states often present in full networks, it is sometimes necessary to construct computationally tractable subnetworks while still carrying sufficient structure for the application at hand—hence, the need for size reducing mappings between PBNs. Construction of mappings to alter PBN structure while at the same time maintaining consistency with the original probability structure have previously been studied [27]. These include projections onto subnetworks. Unfortunately, while projections maintain the probabilistic structure by reducing the number of genes, they also increased the complexity of the Boolean function structure. This paper considers reduction mappings of a PBN that alter the structure of the network while maintaining maximum consistency with the original probability structure. Once this notion of maximum consistency has been defined, the problem reduces to one of optimization. Thus, a key issue to be addressed in this paper is the positing of consistency conditions.

2. DEFINITIONS AND BASIC PROPERTIES

This section provides the definitions and the basic properties of probabilistic Boolean networks as given in [8]. While there have been some generalization of the model [9, 24], we stay with the original definition, as has the original analysis of projection mappings between PBNs [27]—which plays a key role in the present paper. A PBN (V, F, C) is defined by a set of nodes (genes)

$$\begin{aligned} V &= \{x_1, \ldots, x_n\}, \\ x_i &\in \{0, 1\}, \\ i &= 1, \ldots, n, \end{aligned} \tag{1}$$

a list of predictors

$$\begin{aligned} F &= (F_1, \ldots, F_n), \\ F_i &= \left\{f_1^{(i)}, \ldots, f_{l(i)}^{(i)}\right\}, \\ f_j^{(i)} &: \{0, 1\}^n \longrightarrow \{0, 1\}, \end{aligned} \tag{2}$$

and a list

$$\begin{aligned} C &= (C_1, \ldots, C_n), \\ C_i &= \left\{c_1^{(i)}, \ldots, c_{l(i)}^{(i)}\right\} \end{aligned} \tag{3}$$

of selection probabilities $c_j^{(i)} = \Pr\{f^{(i)} = f_j^{(i)}\}$ with respect to a list (vector) of probability distributions $(\nu^{(1)}, \ldots, \nu^{(n)})$, where $\mathbf{f} = (f^{(1)}, \ldots, f^{(n)})$ is a random vector taking values in F. Each node x_i represents the state (expression) of the gene i, where $x_i = 0$ means that the gene i is not expressed and $x_i = 1$ means that it is expressed. Every set F_i contains the possible rules $f_j^{(i)}$ of regulatory interactions for the gene i. These functions are also called *predictors* for the corresponding gene. Updating of the states of all genes in the network is done synchronously according to the functions assigned to the genes, and then the process is repeated. The predictors for every gene x_i are selected simultaneously and randomly (according to the list C) from the sets F_i at every time step.

A *realization* of a PBN is determined at every time step by the vector $\mathbf{f}$. If the predictor for each gene is chosen independently of the other predictors, then the number of all possible realizations $\mathbf{f}_k = (f_{k_1}^{(1)}, \ldots, f_{k_n}^{(n)})$, $k = 1, \ldots, N$, of the PBN is $N = \prod_{j=1}^{n} l(j)$. Even though the domain of every predictor $f_j^{(i)}$ is assumed to be $\{0, 1\}^n$, there are only a few input genes that actually regulate x_i at any given time step. This simplification can be justified by some biological and practical considerations [8]. In general, there is no need of the assumption that $f^{(1)}, f^{(2)}, \ldots, f^{(n)}$ are selected independently; however we make this assumption. A PBN that satisfies this assumption is called *independent*. For an independent PBN, we have

$$P_k = \Pr\{\mathbf{f} = \mathbf{f}_k\} = \prod_{j=1}^{n} \Pr\left\{f^{(i)} = f_{k_j}^{(i)}\right\} = \prod_{j=1}^{n} c_{k_j}^{(i)}. \tag{4}$$

In [8], the list C of selection probabilities is created using the *coefficient of determination* [28, 29].

A PBN can be interpreted as a homogeneous Markov chain relative to the states $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of the network with transition probabilities given by

$$\Pr\{\mathbf{x} \longrightarrow \mathbf{x}'\} = \sum_i P_i, \tag{5}$$

where the summation is over the indices i such that $i : f_{K_{i1}}^{(i)}(x_1, \ldots, x_n) = x_1', \ldots, f_{K_{in}}^{(n)}(x_1, \ldots, x_n) = x_n'$ and K is the matrix with rows given by the possible realizations of the PBN [8].

3. PBN PROJECTION MAPPING

Projection mappings of a PBN A are defined in [27]. They are introduced as an attempt to reduce the complexity of A while maintaining consistency with the original probability structure of the PBN. The basic projection Π_i is a mapping that

transforms the given PBN into a new one, where the number of the genes is reduced by one, that is, the gene x_i in the original network is "deleted." Without loss of generality, we may assume that the deleted gene is the last one, x_n. Thus, $\Pi_n : A \to \hat{A}, \hat{A}(\hat{V}, \hat{F}, \hat{C})$, where

$$\hat{V} = \{x_1, \ldots, x_{n-1}\}, \qquad \hat{F} = (\hat{F}_1, \ldots, \hat{F}_{n-1}), \\ \hat{C} = (\hat{C}_1, \ldots, \hat{C}_{n-1}). \tag{6}$$

Every $\hat{F}_i$ and every $\hat{C}_i$ have twice as many elements as the corresponding sets F_i and C_i in A. Every predictor $f_j^{(i)} \in F_i$ generates two predictors $\hat{f}_{0j}^{(i)}$ and $\hat{f}_{1j}^{(i)}$ according to the rule

$$\hat{f}_{kj}^{(i)}(x_1, \ldots, x_{n-1}) = f_j^{(i)}(x_1, \ldots, x_{n-1}, k), \tag{7}$$

where $k \in \{0, 1\}$ and $(x_1, \ldots, x_{n-1})$ is in $\hat{A}$. The new Boolean functions $\hat{f}_{kj}^{(i)}$, $k = 0, 1$, have transition probabilities

$$\hat{c}_{kj}^{(i)} = c_j^{(i)} \Pr\{x_n = k\}, \quad k \in \{0, 1\}. \tag{8}$$

It is noticed in [27] that there is a difficulty in defining the new selection probabilities $\hat{c}_{kj}^{(i)}$ because the probabilities for the gene x_n depend on the current state probability distribution of the underlying Markov chain. One way to go around the problem is to use the steady state distribution for A or, the stationary distribution for A if there is no steady state distribution. Another way is to estimate $\Pr\{x_n = k\}$, $k = 0, 1$, by running A for some time. In doing so, one has to be aware of the possible transient behavior of those probabilities. Yet, another way to find the values of $\Pr\{x_n = k\}$ is to use the data set from which the original PBN A was created.

4. PBN REDUCTION MAPPINGS

In this paper, we propose a new kind of mapping that also reduces the size of a given PBN. In contrast to the projection mapping discussed in the previous section, this new mapping does not increase the number of the predictors for the genes that remain in the new network. One has to keep in mind that any such mapping might not preserve the probability structure of the original PBN. For example, this will be the case if the deleted gene is *essential* for one of the predictors of the remaining genes [8].

Therefore, the problem is to find a reduction mapping that renders a PBN close to the original one. To be more specific, consider an independent PBN $A(V, F, C)$ and a mapping $\pi_n : A \to \tilde{A}, \tilde{A}(\tilde{V}, \tilde{F}, \tilde{C})$, where

$$\tilde{V} = \{x_1, \ldots, x_{n-1}\}, \qquad \tilde{F} = (\tilde{F}_1, \ldots, \tilde{F}_{n-1}), \\ \hat{C} = (\hat{C}_1, \ldots, \hat{C}_{n-1}), \tag{9}$$

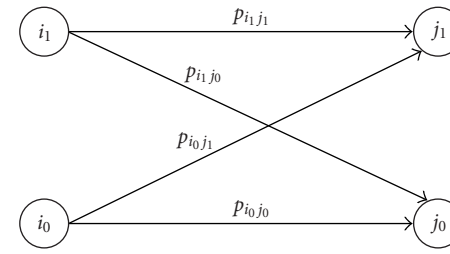
where $\tilde{A}(\tilde{V}, \tilde{F}, \tilde{C})$ is an independent PBN with $\tilde{F}_i = \{\tilde{f}_1^{(i)}, \ldots, \tilde{f}_{l(i)}^{(i)}\}$, $\tilde{f}_j^{(i)} : \{0, 1\}^{n-1} \to \{0, 1\}$, and $\tilde{c}_j^{(i)} = \tilde{\Pr}\{\tilde{f}^{(i)} = \tilde{f}_j^{(i)}\}$ with respect to some probability distribution vector $(\tilde{\nu}^{(1)}, \ldots, \tilde{\nu}^{(n-1)})$. Note that the cardinality of $\tilde{F}_i$ is the same as the cardinality of F_i. The new PBN $\tilde{A}$ is called a *reduced* PBN obtained from the original PBN A by deleting one of the genes in A. As in Section 3, we have assumed without loss of generality that the deleted gene is x_n.

The reduction π_n should yield a PBN that is "close" to the original, and there are various natural ways to interpret this closeness:

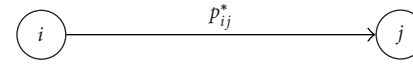
(A) for every $\tilde{c}_j^{(i)}$, $|\tilde{c}_j^{(i)} - c_j^{(i)}| \le \epsilon$ for some given $\epsilon \ge 0$;
(B) the transition probabilities for the state diagrams of A and $\tilde{A}$ are close;
(C) the stationary/steady-state distributions D of A and $\tilde{D}$ of $\tilde{A}$ are close;
(D) every new predictor function $\tilde{f}_j^{(i)}$ is selected as close as possible to both functions $\hat{f}_{kj}^{(i)}$, $k = 0, 1$, given by the projected PBN $\hat{A}$.

Some comments about the preceding conditions are in order.

(A′) In the context of gene regulatory networks, one can expect the number ϵ to be reasonably small, and perhaps even equal to zero, that is, the predictors for the genes in the reduced PBN $\tilde{A}$ have the same selection probabilities as their corresponding predictors from the original PBN A.
(B′) Consider the portion of the state diagram of A containing the states $i_1 = (x_1, \ldots, x_{n-1}, 1)$, $i_0 = (x_1, \ldots, x_{n-1}, 0)$, $j_1 = (x'_1, \ldots, x'_{n-1}, 1)$, and $j_0 = (x'_1, \ldots, x'_{n-1}, 0)$:



where $p_{i_1 j_1}$, $p_{i_0 j_0}$, $p_{i_1 j_0}$, and $p_{i_0 j_1}$ are the corresponding transition probabilities. If one "deletes" the node x_n, this diagram collapses to the following one:



where $i = (x_1, \ldots, x_{n-1})$ and $j = (x'_1, \ldots, x'_{n-1})$ are the corresponding states in $\tilde{A}$, and

$$p_{ij}^* = \Pr\{x_n = 1\}(p_{i_1 j_1} + p_{i_1 j_0}) \\ + \Pr\{x_n = 0\}(p_{i_0 j_1} + p_{i_0 j_0}). \tag{10}$$

The transition probabilities for the reduced PBN $\tilde{A}$ are given by (see [8])

$$\tilde{p}_{ij} = \sum_i \tilde{P}_i, \tag{11}$$

where the summation is over the indices i such that $i : \tilde{f}_{K_{i1}}^{(i)}(x_1, \ldots, x_{n-1}) = x'_1, \ldots, \tilde{f}_{K_{in-1}}^{(i)}(x_1, \ldots, x_{n-1}) = x'_{n-1}$.

Since the transition probability matrices for A and $\tilde{A}$ have different dimensions, one cannot compare them directly. This is why we compare the $\tilde{p}_{ij}$'s to the p^*_{ij}'s, and the term "close" in part (B) refers to the quantity $\max_{i,j}|\tilde{p}_{ij} - p^*_{ij}|$ being small.

(C′) Collapsing the state transition diagram, as described in part (B′), induces a probability distribution D^* on the state space of $\tilde{A}$ in the following way:

$$\begin{aligned}\Pr^*\{\text{state in } \tilde{A} = (x_1,\ldots,x_{n-1})\} &= \Pr\{\text{state in } A = (x_1,\ldots,x_{n-1},0)\} \\ &\quad + \Pr\{\text{state in } A = (x_1,\ldots,x_{n-1},1)\}.\end{aligned} \tag{12}$$

Notice that one cannot compare the distribution D to the distribution $\tilde{D}$ directly because they are defined over different state spaces. This is why the term "close" in (C) refers to the closeness of the distribution D^* and the stationary state distribution $\tilde{D}$ of $\tilde{A}$ in the l_1 sense, that is, to the quantity

$$\|D^* - \tilde{D}\|_{l_1} = \sum_{\mathbf{x}\in\tilde{S}} |\Pr^*\{\mathbf{x}\} - \tilde{\Pr}\{\mathbf{x}\}| \tag{13}$$

being small. Here, $\tilde{S} = \{0,1\}^{n-1}$ is the set of all states in $\tilde{A}$, and $\tilde{\Pr}$ is associated with $\tilde{D}$.

(D′) Using the notation from (D), we have the following proposition.

Proposition 1. *Given a PBN A with a stationary state distribution D, consider the projected PBN $\hat{A}$. Then*

$$E_D\left(\frac{\partial f_j^{(i)}}{\partial x_n}\right) = \left\|\hat{f}_{0j}^{(i)} - \hat{f}_{1j}^{(i)}\right\|_{l_1^{2^{n-1}}} = E_{D^*}\left(\left|\hat{f}_{0j}^{(i)} - \hat{f}_{1j}^{(i)}\right|\right). \tag{14}$$

Here the space $l_1^{2^{n-1}}$ is endowed with the probability measure $\tilde{\Pr}$ defined by the distribution D^*, and E_D means the expectation of the corresponding random variable with respect to the distribution D.

Proof. The claim in this proposition becomes obvious if one notices that for every state $(x_1,\ldots,x_{n-1},x_n) \in \{0,1\}^n$, where $\partial f_j^{(i)}/\partial x_n = 1$, there are two terms in the sum that compute $E_D(\partial f_j^{(i)}/\partial x_n)$, namely, $\Pr\{(x_1,\ldots,x_{n-1},0)\}$ and $\Pr\{(x_1,\ldots,x_{n-1},1)\}$. □

The proposition plays an important role in selecting the new predictor function $\tilde{f}_j^{(i)}$. Notice that the expectation $E_D(\partial f_j^{(i)}/\partial x_n)$ represents the influence $I_n(f_j^{(i)})$ of the gene x_n on the predictor $f_j^{(i)}$ (cf. [8]). In the special case when x_n is not essential for the function $f_j^{(i)}$, the new predictor can be selected to be identically equal to either of the two possible predictors $\hat{f}_{0j}^{(i)}$ and $\hat{f}_{1j}^{(i)}$ in the projected PBN $\hat{A}$. Generally speaking, the selection of the new predictor $\tilde{f}_j^{(i)}$ should minimize both $E_{D^*}(|\hat{f}_{0j}^{(i)} - \tilde{f}_j^{(i)}|)$ and $E_{D^*}(|\tilde{f}_j^{(i)} - \hat{f}_{1j}^{(i)}|)$. The inequality

$$\begin{aligned}E_{D^*}\left(\left|\hat{f}_{0j}^{(i)} - \hat{f}_{1j}^{(i)}\right|\right) &\le E_{D^*}\left(\left|\hat{f}_{0j}^{(i)} - \tilde{f}_j^{(i)}\right|\right) \\ &\quad + E_{D^*}\left(\left|\tilde{f}_j^{(i)} - \hat{f}_{1j}^{(i)}\right|\right)\end{aligned} \tag{15}$$

provides a measurement of how well the reduction mapping preserves the predictors from the original PBN. "Deleting" a gene x_k with bigger influence $I_k(f_j^{(i)})$ on the predictor $f_j^{(i)}$ produces a new predictor $\tilde{f}_j^{(i)}$ which cannot be closer to $f_j^{(i)}$ when compared to the new predictor resulting from the "deletion" of a gene x_l with smaller influence $I_l(f_j^{(i)})$ on $f_j^{(i)}$. In other words, "deleting" essential genes from the original PBN comes with a "price"—the predictor functions for the reduced PBN cannot be too close to the original predictors.

The selection of every function $\tilde{f}_j^{(i)} \in \tilde{F}_i$ has to be performed pointwise, that is, for each state in $\tilde{S}$, define

$$U = \left\{\mathbf{x} = (x_1,\ldots,x_{n-1}) \in \tilde{S} : \hat{f}_{0j}^{(i)}(\mathbf{x}) = \hat{f}_{1j}^{(i)}(\mathbf{x})\right\} \tag{16}$$

and $W = \tilde{S} \setminus U$. Clearly, $\tilde{f}_j^{(i)} \equiv \hat{f}_{0j}^{(i)} \equiv \hat{f}_{1j}^{(i)}$ on the set U. For the states in the remaining set W, one has to decide to what degree one favors certain states in $S = \{0,1\}^n$ which in its turn defines $\tilde{f}_j^{(i)}$ as either equal to $\hat{f}_{0j}^{(i)}$ or to $\hat{f}_{1j}^{(i)}$. Motivated by the preceding remarks about the conditions (A), (B), (C), and (D), we now design a selection procedure for the functions $\tilde{f}_j^{(i)}$.

Selection procedure

(a) For all i, j, select numbers $-1 \le \omega_j^{(i)} \le 1$.

(b) For every state $\mathbf{x} = (x_1,\ldots,x_{n-1}) \in W$, define

$$\tilde{f}_j^{(i)}(\mathbf{x}) = \begin{cases} \hat{f}_{0j}^{(i)}(\mathbf{x}) & \text{if } \Pr\{(x_1,\ldots,x_{n-1},0)\} \\ & \quad > \omega_j^{(i)} + \Pr\{(x_1,\ldots,x_{n-1},1)\}; \\ \hat{f}_{1j}^{(i)}(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{17}$$

(c) For every state $\mathbf{x} = (x_1,\ldots,x_{n-1}) \in U$, set $\tilde{f}_j^{(i)}(\mathbf{x}) = \hat{f}_{1j}^{(i)}(\mathbf{x})$.

Notice that the condition on the numbers $\omega_j^{(i)}$ is natural since we are dealing with probabilities.

Our selection procedure leads to the following optimization problem.

Problem 1. Find $\tilde{F}$ that achieves $\min_\Omega \max_{i,j}|\tilde{p}_{ij} - p^*_{ij}|$ subject to

(i) $\tilde{c}_j^{(i)} = c_j^{(i)}$, $1 \le i \le n-1$, $1 \le j \le l(n)$,

(ii) $\Omega = \{\omega_j^{(i)} : -1 \le \omega_j^{(i)} \le 1, 1 \le i \le n-1, 1 \le j \le l(n)\}$.

Remark 1. The above problem has a solution: it is enough to notice that Ω is a compact set.

Remark 2. From a computational point of view, the only values for $\omega_j^{(i)}$ one should consider are the differences

$\Pr\{(x_1,\ldots,x_{n-1},0)\} - \Pr\{(x_1,\ldots,x_{n-1},1)\}$. This essentially reduces Ω to a finite set. Notice that if all $\omega_j^{(i)} \equiv -1$, then $\tilde{f}_j^{(i)}(\mathbf{x}) \equiv \hat{f}_{0j}^{(i)}(\mathbf{x})$. The other extreme choice is when all $\omega_j^{(i)} \equiv 1$ which produces $\tilde{f}_j^{(i)}(\mathbf{x}) \equiv \hat{f}_{1j}^{(i)}(\mathbf{x})$. One can see that different choices for the $\omega_j^{(i)}$'s could be based on how much one favors certain states in the original PBN. In the following simulations, we always set $\omega_j^{(i)} = 0$ which means that we do not assume any additional information, and the selection of the new predictor functions is based only on the probability distribution of the states of A.

5. COMPARISON BETWEEN THE PROJECTION AND THE REDUCTION MAPS

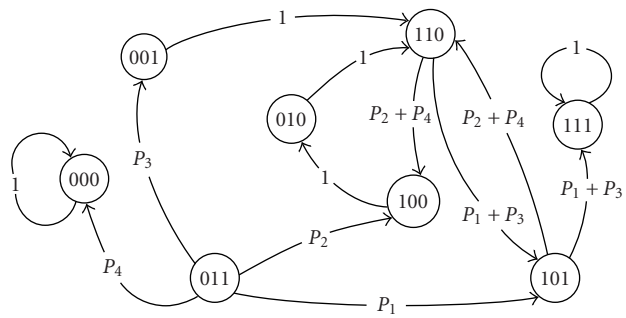
One should immediately notice the difference in defining the reduction and the projection mappings. While the projection is based on the probability distribution of a single gene, the reduction mapping is defined using the probability distribution of the entire collection of states of the given PBN. To illustrate this difference, we consider one particular example of a PBN (cf. [8]).

Example 1. Let $A(V,F,C)$ be a PBN consisting of three genes $V = \{x_1,x_2,x_3\}$ and function sets $F = (F_1,F_2,F_3)$, where $F_1 = \{f_1^{(1)}, f_2^{(1)}\}$, $F_2 = \{f_1^{(2)}\}$, and $F_3 = \{f_1^{(3)}, f_2^{(3)}\}$, and the predictor functions are given by the truth table (Table 1).

Table 1

$x_1x_2x_3$	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
000	0	0	0	0	0
001	1	1	1	0	0
010	1	1	1	0	0
011	1	0	0	1	0
100	0	0	1	0	0
101	1	1	1	1	0
110	1	1	0	1	0
111	1	1	1	1	1
$c_j^{(i)}$	0.6	0.4	1	0.5	0.5

After computing the transition probabilities, (cf. [8]), we arrive at the following directed graph/state transition diagram:



Here, $P_1 = 0.3$, $P_2 = 0.3$, $P_3 = 0.2$, and $P_4 = 0.2$. Next, we start with a uniform state probability distribution $D_{in} = \{1/8, 1/8, \ldots, 1/8\}$ for the states in the state space S of A, and then run the corresponding Markov chain for some large number of iterations. Notice that even if the given network does not possess a steady state distribution, the result after running the Markov chain sufficiently long time is approximately the stationary state distribution D that corresponds to D_{in}. The simulation gives $D = \{0.15, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.85\}$. Using this distribution, one can compute the projected $\hat{A}$ and the reduced $\tilde{A}$ networks, as well as their probability transition matrices. After running the Markov processes associated with these two transition probability matrixes, we obtain the stationary state distributions $\hat{D}$ for $\hat{A}$, and $\tilde{D}$ for $\tilde{A}$. For the case when the deleted gene is x_3, we get

$$\begin{aligned} \hat{D} &= \{0.006734, 0.022778, 0.138384, 0.831647\}, \\ \tilde{D} &= \{0.35, 0.0, 0.0, 0.65\}. \end{aligned} \tag{18}$$

The stationary state distribution for the transition probability matrix $(p_{i,j}^*)_{i=1,\,j=1}^{4,4}$, produced after the collapsing procedure described in part (D) (Section 4), is $D_1 = \{0.008145, 0.020362, 0.135747, 0.835747\}$. One can notice the similarity between D_1 and $\hat{D}$ and their apparent difference from $\tilde{D}$. At the same time, the distribution $D^* = \{0.15, 0.0, 0.0, 0.85\}$ described in part (D), Section 4, is similar to $\tilde{D}$. This should not be surprising—both the projection and the "collapsing" mappings are based on the probability distribution of a single gene, x_3 in our example, while the reduction mapping is based on the probability distribution of the entire collection of states in the original PBN. Thus the optimization criterion described in Problem 1 becomes a natural compromise between these two possible approaches of reducing the original PBN size.

Inequality (15) can be used in deciding which gene, after being eliminated from the network, will have a minimal impact on the stationary distribution of the original PBN. Since the left-hand side of (15) represents the influence $I_n(f_j^{(i)})$ of a gene x_n on the predictor $f_j^{(i)}$, one can say that, in general, deleting genes with smaller influences on the remaining predictors will result in a better chance of preserving the stationary state distribution of the original PBN. Here, we provide the values for the influences of x_3 on the remaining predictors and then two more simulations for the same example, where the other two possible genes x_2 and x_1 are deleted from the original PBN. The influences of x_3 on the remaining predictors are $I_3(f_1^{(1)}) = 0.15$, $I_3(f_2^{(1)}) = 0.15$, and $I_3(f_1^{(2)}) = 1$. After deleting x_2 from A, the corresponding stationary state distribution is $\tilde{D} = \{0.25, 0.0, 0.0, 0.75\}$, and the influences of x_2 on the remaining predictors are $I_2(f_1^{(1)}) = 0.15$, $I_2(f_2^{(1)}) = 0.15$, $I_2(f_3^{(1)}) = 0$, and $I_2(f_3^{(2)}) = 0.85$. After deleting x_1 from A, the corresponding stationary state distribution is $\tilde{D} = \{0.5, 0.0, 0.0, 0.5\}$, and the influences of x_1 on the remaining predictors are $I_1(f_2^{(1)}) = 1$, $I_1(f_3^{(1)}) = 0$, and $I_1(f_3^{(2)}) = 0.85$.

It appears that the gene x_1 with the biggest total influence distorts the stationary state distribution the most but one should be careful when generalizing this observation. Gene influences can be computed based on different probability distributions (cf. [8]). In addition, deleting different genes from the original PBN results in reduced PBNs with different state spaces. Finally, the left-hand side of (15) is just a lower bound that governs the selection procedure in constructing $\tilde{A}$, and that the lower bound might not be achieved during the selection procedure.

6. SIMULATION RESULTS

The reduction mapping has been tested using coefficient of determination (COD) microarray data for a network A consisting of 10 genes [23]. The genes of interest in the network are *PIRIN, WNT5A, S100P, RET-1, MMP-3, PHO-C, STC2, MART-1, HADHB*, and *SYNUCLEIN*. The network is reduced down to 7 genes by subsequently deleting the last three genes, starting with *SYNUCLEIN*. Table 2 presents lists of some of the states in the stationary/steady distributions for the full network A and the reduced networks $\tilde{A}_{10}$, $\tilde{A}_{10,9}$, and $\tilde{A}_{10,9,8}$, where the indices indicate which genes in A are deleted. For example, $\tilde{A}_{10,9,8}$ is the reduced network after deleting the genes *SYNUCLEIN*, *HADHB*, and *MART-1*. The states are presented by binary strings of ten digits, where 0 indicates that the corresponding gene is "OFF" and 1 indicates that the corresponding gene is "ON." The leftmost digit represents *PIRIN* and then the remaining digits represent the following genes in the network with the rightmost digit representing *SYNUCLEIN*. Next to every given state, its corresponding weight in the stationary state distribution of the network is given. Only states with weight bigger than 0.0001 are shown.

One can notice the presence of a very "heavy" state, 1010000111, in the stationary/steady state distribution of the full network. That is in agreement with the COD data set, where the same state is present in 8 out of 31 samples (see [23] for a related discussion). The reduction mapping maintains the structure of the stationary state distribution of the full network, specifically, the states 101000011, 10100001, and 1010000 carry most of the weight in the stationary/steady state distributions of their corresponding reduced networks.

7. CONCLUSION

The new mapping introduced in this paper offers a way of reducing the size of a given PBN by using the stationary probability distribution on the state space of the PBN. At the same time, it minimizes the distance between the reduced network and the projected PBN introduced in [27]. The distance is given in terms of the distance between their corresponding probability transition matrices. One should notice that the construction of the projected PBN is based on the probability distribution of a single gene, and that the same single gene probability distribution could happen under many different stationary distributions on the state space of the original PBN.

Table 2

For the full network A:
(0000111000, 0.003773); (0000111001, 0.003117);
(0001111000, 0.001905); (0010000111, 0.001715);
(0100011000, 0.001030); (0100101000, 0.010710);
(0100101001, 0.012694); (0100111000, 0.023957);
(0100111001, 0.026482); (0101101000, 0.004985);
(0101101001, 0.001685); (0101111000, 0.011730);
(0101111001, 0.003710); (0110111001, 0.001352);
(0111111000, 0.001416); (1010000101, 0.002299);
(1010000111, 0.832929)
For $\tilde{A}_{10}$:
(000011100, 0.010795); (000111100, 0.002513);
(001011100, 0.001944); (010010100, 0.140539);
(010011100, 0.083694); (010110100, 0.020368);
(010111000, 0.001241); (010111100, 0.014547);
(011011000, 0.001116); (011011100, 0.005920);
(011111000, 0.001743); (011111100, 0.003310);
(101000010, 0.001003); (101000011, 0.689413)
For $\tilde{A}_{10,9}$:
(00000110, 0.001528); (00001110, 0.017951);
(00011110, 0.004141); (00101110, 0.003209);
(01000110, 0.001423); (01001010, 0.230668);
(01001100, 0.001481); (01001110, 0.137728);
(01011010, 0.033485); (01011100, 0.002186);
(01011110, 0.024005); (01101100, 0.001911);
(01101110, 0.009774); (01111100, 0.003293);
(01111110, 0.005520); (10100001, 0.499967)
For $\tilde{A}_{10,9,8}$:
(0000011, 0.001523); (0000111, 0.020527);
(0001110, 0.001065); (0001111, 0.004695);
(0010111, 0.003700); (0100011, 0.001814);
(0100101, 0.269151); (0100110, 0.001628);
(0100111, 0.160629); (0101101, 0.039116);
(0101110, 0.002575); (0101111, 0.027858);
(0110110, 0.002169); (0110111, 0.011428);
(0111110, 0.004034); (0111111, 0.006365);
(1010000, 0.429211)

ACKNOWLEDGMENTS

This research was supported by the National Cancer Institute (CA90301), the National Human Genome Research Institute, and the Translational Genomics Research Institute.

REFERENCES

[1] L. Glass and S. A. Kauffman, "The logical analysis of continuous, nonlinear biochemical control networks," *Journal of Theoretical Biology*, vol. 39, no. 1, pp. 103–129, 1973.

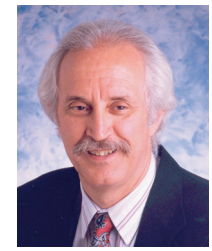
[2] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[3] S. A. Kauffman, *Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, NY, USA, 1993.

[4] R. Somogyi and C. Sniegoski, "Modeling the complexity of

genetic networks: understanding multigenic and pleiotropic regulation," *Complexity*, vol. 1, no. 6, pp. 45–63, 1996.
[5] Z. Szallasi and S. Liang, "Modeling the normal and neoplastic cell cycle with "realistic Boolean genetic networks": their application for understanding carcinogenesis and assessing therapeutic strategies," in *Proc. Pacific Symposium on Biocomputing (PSB '98)*, vol. 3, pp. 66–76, Maui, Hawaii, USA, January 1998.
[6] R. Thomas, D. Thieffry, and M. Kaufman, "Dynamical behavior of biological regulatory networks. I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state," *Bulletin of Mathematical Biology*, vol. 57, no. 2, pp. 247–276, 1995.
[7] A. Wuensche, "Genomic regulation modeled as a network with basins of attraction," in *Proc. Pacific Symposium on Biocomputing (PSB '98)*, vol. 3, pp. 89–102, Maui, Hawaii, USA, January 1998.
[8] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.
[9] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.
[10] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3/4, pp. 601–620, 2000.
[11] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young, "Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks," in *Proc. Pacific Symposium on Biocomputing (PSB '01)*, pp. 422–433, Honolulu, Hawaii, USA, January 2001.
[12] E. J. Moler, D. C. Radisky, and I. S. Mian, "Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in S. cerevisiae," *Physiological Genomics*, vol. 4, no. 2, pp. 127–135, 2000.
[13] K. Murphy and I. S. Mian, "Modeling gene expression data using dynamic Bayesian networks," Tech. Rep., University of California, Berkeley, Calif, USA, 1999.
[14] A. Arkin, J. Ross, and H. H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells," *Genetics*, vol. 149, no. 4, pp. 1633–1648, 1998.
[15] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: in numero molecular biology," *Nature Reviews Genetics*, vol. 2, no. 4, pp. 268–279, 2001.
[16] P. Smolen, D. Baxter, and J. Byrne, "Mathematical modeling of gene networks," *Neuron*, vol. 26, no. 3, pp. 567–580, 2000.
[17] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery," *Journal of Molecular Medicine*, vol. 77, no. 6, pp. 469–480, 1999.
[18] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions," in *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98)*, pp. 695–702, San Francisco, Calif, USA, January 1998.
[19] T. Akutsu, S Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, no. 8, pp. 727–734, 2000.
[20] P. D'Haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.
[21] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," in *Proc. Pacific Symposium on Biocomputing (PSB '98)*, vol. 3, pp. 18–29, Maui, Hawaii, USA, January 1998.
[22] X. Zhou, X. Wang, and E. R. Dougherty, "Construction of genomic networks using mutual-information clustering and reversible-jump Markov-Chain-Monte-Carlo predictor design," *Signal Processing*, vol. 83, no. 4, pp. 745–761, 2003.
[23] S. Kim, H. Li, E. R. Dougherty, et al., "Can Markov chain models mimic biological regulation?," *Journal of Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.
[24] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.
[25] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," *Journal of Biological Systems*, vol. 10, no. 4, pp. 431–445, 2002.
[26] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, no. 1/2, pp. 169–191, 2003.
[27] E. R. Dougherty and I. Shmulevich, "Mappings between probabilistic Boolean networks," *Signal Processing*, vol. 83, no. 4, pp. 799–809, 2003.
[28] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.
[29] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *Journal of Biomedical Optics*, vol. 5, no. 4, pp. 411–424, 2000.

Ivan Ivanov is a Visiting Research Scholar in the Department of Electrical Engineering at Texas A&M University in College Station and a trainee in the training program in bioinformatics at the Department of Statistics at Texas A&M University. He received his M.S. degree in mathematics modeling from Sofia University St. Kilmet Ohridski in 1987 and his Ph.D. degree in mathematics from the University of South Florida in 1999. His current research focuses on genomic signal processing and, in particular, on modeling the genomic regulatory mechanisms. He is working in the Genomic Signal Processing Laboratory at Texas A&M University.



Edward R. Dougherty is a Professor in the Department of Electrical Engineering at Texas A&M University in College Station. He holds an M.S. degree in computer science from Stevens Institute of Technology in 1986 and a Ph.D. degree in mathematics from Rutgers University in 1974. He is the author of eleven books and the editor of other four books. He has published more than one hundred journal papers, is an SPIE Fellow, and has served as an Editor of the Journal of Electronic Imaging for six years. He is currently Chair of the SIAM Activity Group on Imaging Science. Prof. Dougherty has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His current research focuses on genomic signal processing, with the central goal being to model genomic regulatory mechanisms. He is Head of the Genomic Signal Processing Laboratory at Texas A&M University.

EURASIP Journal on Applied Signal Processing 2004:1, 132–137

Genomic Signals of Reoriented ORFs

Paul Dan Cristea

Biomedical Engineering Center, Politehnica University of Bucharest, Splaiul Independentei 313, Bucharest 77206, Romania
Email: pcristea@dsp.pub.ro

Received 14 March 2003; Revised 12 September 2003

Complex representation of nucleotides is used to convert DNA sequences into complex digital genomic signals. The analysis of the cumulated phase and unwrapped phase of DNA genomic signals reveals large-scale features of eukaryote and prokaryote chromosomes that result from statistical regularities of base and base-pair distributions along DNA strands. By reorienting the chromosome coding regions, a "hidden" linear variation of the cumulated phase has been revealed, along with the conspicuous almost linear variation of the unwrapped phase. A model of chromosome longitudinal structure is inferred on these bases.

Keywords and phrases: genomic signals, open reading frames, ORF orientation.

1. INTRODUCTION

The conversion of nucleotide sequences into digital signals offers the opportunity to apply signal processing methods to analyze genomic information. Using the genomic signal approach, long-range features of DNA sequences, maintained over distances of 10^6–10^8 base pairs, that is, at the scale of whole chromosomes, have been found [1, 2, 3, 4, 5, 6, 7]. One of the most conspicuous results is that the unwrapped phase of the complex genomic signal varies almost linearly along all investigated chromosomes for both prokaryotes and eukaryotes. The slope is specific for various taxa and chromosomes. Such a behavior reveals a large-scale regularity in the distribution of the pairs of successive nucleotides—a rule for the statistics of second order: *the difference between the frequency of positive nucleotide-to-nucleotide transitions (*A → G, G → C, C → T, T → A*) and that of negative transitions (the opposite ones) along a strand of nucleic acid tends to be small, constant, and taxon and chromosome specific*. There is a similarity between this rule and Chargaff's rules referring to the frequencies of occurrence of nucleotides, that is, to statistics of the first order [8].

The paper shows that the abrupt changes in nucleotide frequencies along DNA strands of prokaryote chromosomes, as revealed by the piecewise linear variation of the cumulated phase of complex genomic signals [1, 2, 3, 4, 5, 6, 7] or by the skew diagrams [9, 10, 11], are the effect of corresponding abrupt changes in the distribution of direct and inverse open reading frames (ORFs) along the strand. It is also shown that, by reorienting all the negative (inverse) ORFs in the direction of the positive (direct) ones, an almost linear variation of the cumulated phase along the concatenated sequence is obtained, corresponding to almost constant frequencies of nucleotides along the entire chain of concatenated reordered ORFs. This large-scale homogeny of the reordered ORFs, together with the taxon specific large-scale regularities of the actual nucleic DNA strands, suggests that the distribution of direct and inverse coding segments along chromosomes, as reflected in the slope of the cumulated phase, has a functional role, most probably linked to the control of the crossing-over/recombination process, thus playing a role in the separation of species. A similar property probably exists in eukaryote chromosomes too, but the relative extension of the coding regions is much lower than in the case of prokaryotes, so that there is too little information for the reordering of the extremely large number of direct and inverse individual chromosome patches.

The paper also presents a model of chromosome longitudinal structure. The model explains why the frequency of nucleotide-to-nucleotide transitions does not change significantly in the points of abrupt changes of the nucleotide frequencies or as a consequence of ORF reordering. Correspondingly, the model explains the ubiquitous almost linear variation of the unwrapped phase of the genomic signals along all investigated chromosomes.

2. DATA AND METHOD

Complete genomes or complete sets of available contigs for eukaryote and prokaryote taxa have been downloaded from the GenBank [12] database of National Institutes of Health (NIH), converted into genomic signals, and analyzed at the scale of whole chromosomes.

As the detailed methodology of the nucleotide, codon, and amino acid sequence conversion into digital signals has been presented elsewhere [3, 4], we give here only a short summary of the quadrantal complex representation used throughout this paper. The nucleotides (adenine (A), cytosine (C), guanine (G), and thymine (T)) are mapped to four

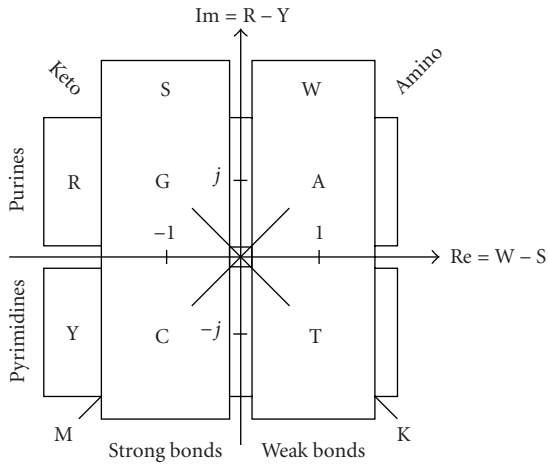


Figure 1: Nucleotide quadrantal complex representation.

complex numbers as shown in Figure 1:

$$a = 1 + j, \quad c = -1 - j, \quad g = -1 + j, \quad t = 1 - j. \tag{1}$$

The representation (1) conserves the main six classes of nucleotides:

(i) strong bonds S = {C, G},
(ii) weak bonds W = {A, T},
(iii) amino M = {A, C},
(iv) keto K = {G, T},
(v) pyrimidines Y = {C, T},
(vi) purines R = {A, G},

and readily expresses the W-S and R-Y dichotomies. This representation allows also the classification of nucleotide pairs in three sets of transitions, in accordance with the change of the unwrapped phase they produce when occurring in a sequence:

(i) the *positive transitions* A → G, G → C, C → T, and T → A that determine a variation with $+\pi/2$ in the trigonometric sense,
(ii) the set of *negative transitions* A → T, T → C, C → G, and G → A—that determines a variation of $-\pi/2$, clockwise,
(iii) the set of *neutral transitions* that correspond to a zero-mean change of the unwrapped phase.

The slopes s_c of the cumulated phase and s_u of the unwrapped phase of a complex genomic signal, obtained by applying the representation (1) to a DNA sequence, are linked to the nucleotide and the nucleotide-to-nucleotide transition frequencies by the following equations [2]:

$$s_c = \frac{\pi}{4}\left[3(f_G - f_C) + (f_A - f_T)\right], \tag{2}$$

$$s_u = \frac{\pi}{2}(f_+ - f_-), \tag{3}$$

where f_A, f_C, f_G, and f_T are the nucleotide frequencies, while f_+ and f_- are the positive and negative transition frequencies.

Thus, the phase analysis of complex genomic signals is able to reveal features of both the nucleotide frequencies and the nucleotide-to-nucleotide transition frequencies along DNA strands.

Relations (1) can be seen as representing the nucleotides in two orthogonal bipolar binary systems with complex bases (units).

3. A MODEL OF DNA LONGITUDINAL STRUCTURE

The chromosomes of both prokaryotes and eukaryotes have a very "patchy" structure comprising many intertwined coding and noncoding segments oriented in a direct and inverse sense. The reversed orientation of DNA segments has been found first for the coding regions, where direct and inverse ORFs have been identified. The analysis of the modalities in which DNA segments can be chained together along the DNA double helix is important for understanding genomic signal large-scale properties [1, 2, 3].

The direction reversal of a DNA segment is always accompanied by the switching of the antiparallel strands of its double helix. This property is a direct result of the requirement that all the nucleotides be linked to each other along the DNA strands only in the 5′ to 3′ sense.

Figure 2 schematically shows the way in which the 5′ to 3′ orientation restriction is satisfied when a segment of a DNA double helix is reversed and/or has its strands switched. In the case in Figure 2a, the two component helices have the chains $(A_0A_1)(A_1A_2)(A_2A_3)$ and $(B_0B_1)(B_1B_2)(B_2B_3)$, respectively, ordered in the 5′ to 3′ sense indicated by the arrows. The reversal of the middle segment, without the corresponding switching of its strands (Figure 2b), would generate the forbidden chains $(A_0A_1)(A_2A_1)(A_2A_3)$ and $(B_0B_1)(B_2B_1)(B_2B_3)$ that violate the 5′ to 3′ alignment condition. Similarly, the switching of the strands of the middle segment, without its reversal, would generate the equally forbidden chains $(A_0A_1)(B_2B_1)(A_2A_3)$ and $(B_0B_1)(A_2A_1)(B_2B_3)$ shown in Figure 2c. Finally, the conjoint reversal of the middle segment and the switching of its strands (Figure 2d) generate the chains $(A_0A_1)(B_1B_2)(A_2A_3)$ and $(B_0B_1)(A_1A_2)(B_2B_3)$, compatible with the 5′ to 3′ orientation condition. As a consequence, there is always a pair of changes (direction reversal and strand switching) produced by an inversed insertion of a DNA segment so that the sense/antisense orientation of individual DNA segments affects the nucleotide frequencies but not the frequencies of the positive and negative transitions. Figure 3 shows the effect of the *segment reversal* and *strand switching* transformations on the positive and negative nucleotide-to-nucleotide transitions for the case of the complex genomic signal representation given by (1). After a pair of segment reversal and strand switching transformations of a DNA segment, the nucleotide transitions do not change their type (positive or negative). As a consequence, the slope of the unwrapped phase does not change as the slope of the cumulated phase. This explains why the cumulated phase and the unwrapped phase

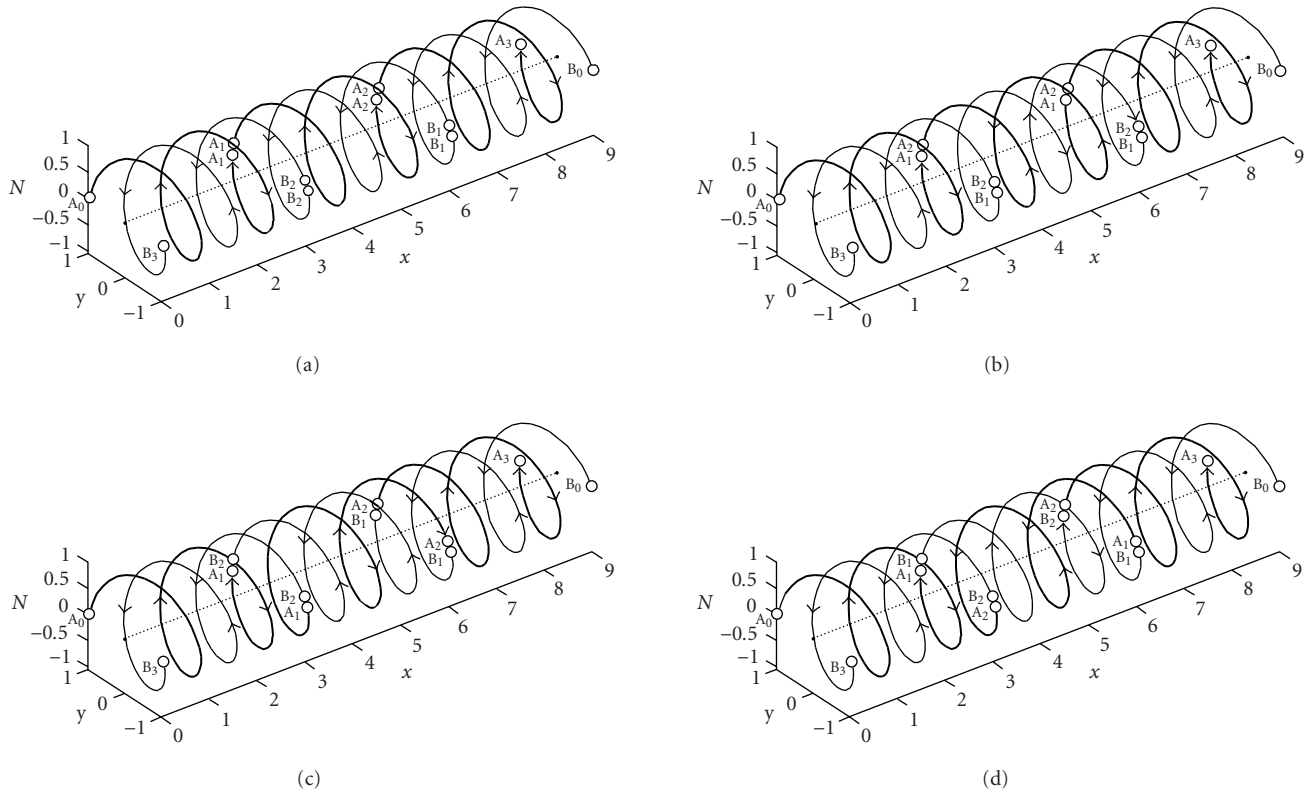


Figure 2: Schematic representations of the direction reversal of a DNA segment. (a) Initial state in which the two antiparallel strands have all the marked segments ordered in the 5′ to 3′ direction, indicated by arrows. (b) Hypothetic reversal of the middle segment without the switching of the strands. (c) Hypothetic switching of strands for the middle segment without its reversal. (d) Direction reversal and strand switching for the middle segment. The 5′ to 3′ alignment condition is violated in cases (b) and (c) but reestablished in (d).

of genetic signals have completely different types of variations along DNA molecules that contain a large number of reversed segments.

4. CUMULATED AND UNWRAPPED PHASE VARIATION ALONG CHROMOSOMES AND CONCATENATED REORIENTED CODING REGIONS

Figure 4 presents the cumulated and the unwrapped phases of the complete circular chromosome of *Salmonella typhi*, the multiple-drug resistant strain CT18 [13] (accession AL5113382 [12]). The locations of the breaking points, where the cumulated phase changes the sign of the slope of its variation along the DNA strand, are given in Figure 4. Even if, locally, the cumulated phase and the unwrapped phase do not have a smooth variation, at the large scale used in Figure 4, the variation is quite smooth and regular. A pixel in the curves of Figure 4 represents 6050 data points, but the absolute value of the difference between the maximum and minimum values of the data in the set of points represented by each pixel is smaller than the vertical pixel dimension expressed in data units. This means that the local data variation falls between the limits of the width of the line used for the plot so that the graphic representation of data by a line is adequate. As found for other prokaryotes [2, 3, 4, 5], the cumulated phase has an approximately piecewise linear variation over two almost equal domains, one of positive slope

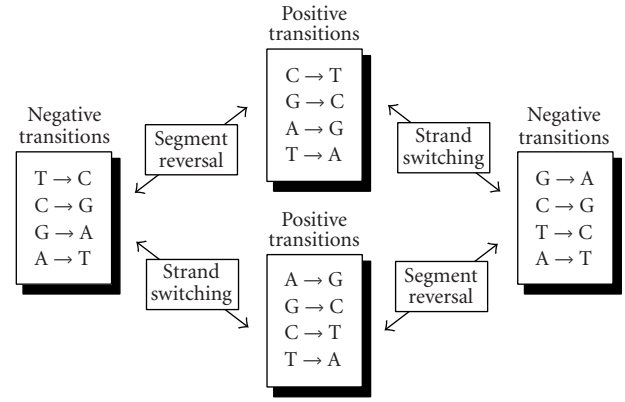


Figure 3: Effect of segment reversal and strand switching on positive and negative nucleotide-to-nucleotide transitions. An even number of transforms do not change the type of the transitions.

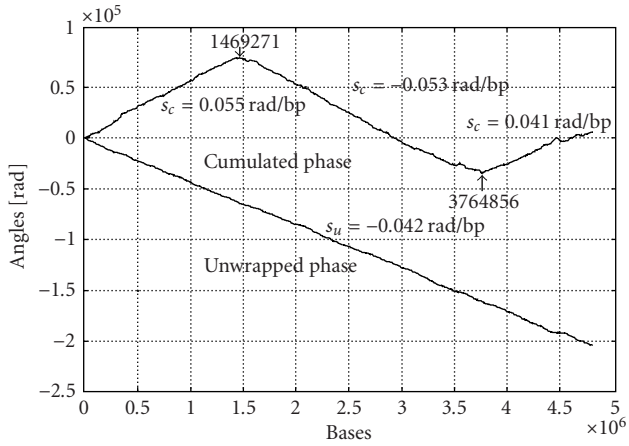


Figure 4: Cumulated and unwrapped phases for the genomic signal of the complete chromosome (4809037 bp) of *Salmonella typhi* [13] (accession AL5113382 [12]).

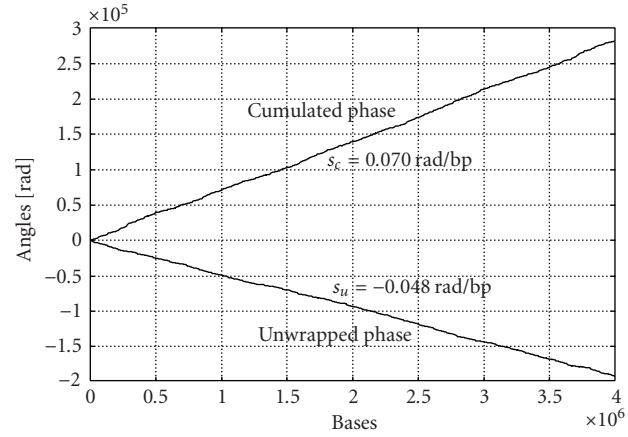


Figure 5: Cumulated and unwrapped phases of the genomic signal for the concatenated 4393 reoriented coding regions (3999478 pb) of *Salmonella typhi* genome [13] (accession AL5113382 [12]).

(apparently divided in the intervals 1-1469271 and 3764857-4809037, but actually contiguous on the circular chromosome) and the second of negative slope (1469272-3764856), while the unwrapped phase has an almost linear variation for the entire chromosome, showing little or no change in the breaking points. The breaking points, like the extremes of the integrated skew diagrams, have been put in relation with the origins and termini of chromosome replichores [2, 9, 11]. The slope of the cumulated phase in each domain is related to the nucleotide frequency in that domain by (2). In the breaking points, a macroswitching of the strands, accompanied by a reversal of one of the domain-large segments, occurs. On the other hand, the two domains comprise a large number of much smaller segments, oriented in the direct and the inverse sense. At the junctions of these segments, reversals and switchings of DNA helix segments take place as described in Section 3. The average slope of each large domain is actually determined by the density of direct and inverse small segments along that domain. This model can be verified by using the "**.ffn*" files in the GenBank [12] database that contain the coding regions of the sequenced genomes, together with their orientation. Concatenating the coding regions oriented in the positive direction (positive ORFs) with the reoriented (reversed and complemented) coding regions read in the negative direction (negative ORFs), a nucleotide sequence with all the coding regions (exons and introns) oriented in the same direction is obtained. Because the intergenic regions for which the orientation is not known have to be left out of the reoriented sequence, this new sequence is shorter than the one that contains the entire chromosome or all the available contigs given in the "**.gbk*" files of the GenBank database [12].

Figure 5 shows the cumulated and unwrapped phases of the genomic signal obtained by concatenating the 4393 reoriented coding regions of *Salmonella typhi* genome [13] (accession AL5113382 [12]). Each inverse coding region (inverse ORF) has been reversed and complemented, that is,

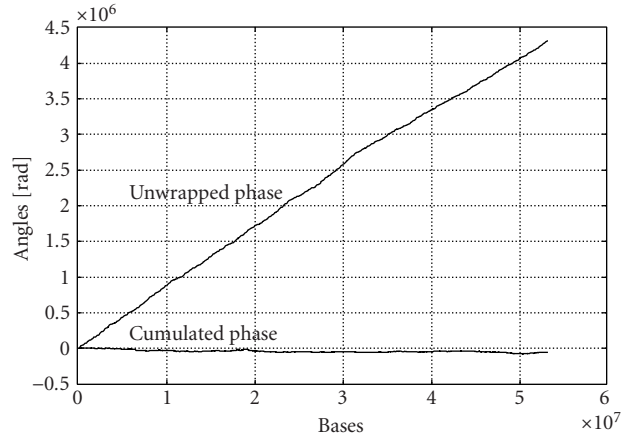


Figure 6: Cumulated and unwrapped phases along the complete chromosome 4 of *Mus musculus* [14] (NT019246 53208110 bp [12]).

the nucleotides inside the same W (adenine-thymine) or S (cytosine-guanine) class have been replaced with each other to take into account the switching of the strands that accompanies the segment reversal.

As expected from the model, the breaking points in the cumulated phase disappear and the absolute values of the slopes increase as there is no longer interweaving of direct and inverse ORFs. The average slope s_c of the cumulated phase of a genomic signal for a domain is linked to the average slope $s_c^{(0)}$ of the concatenated reoriented coding regions by the relation

$$s_c = \frac{\sum_{k=1}^{n_+} l_k^{(+)} - \sum_{k=1}^{n_-} l_k^{(-)}}{\sum_{k=1}^{n_+} l_k^{(+)} + \sum_{k=1}^{n_-} l_k^{(-)}} s_c^{(0)}, \tag{4}$$

where $\sum_{k=1}^{n_+} l_k^{(+)}$ and $\sum_{k=1}^{n_-} l_k^{(-)}$ are the total lengths of the n^+ direct and n^- inverse ORFs in the given domain.

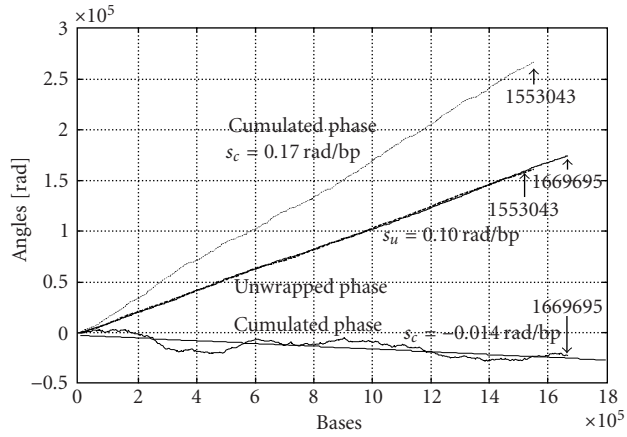


Figure 7: Cumulated and unwrapped phases of the genomic signals for the complete nucleotide sequence and the concatenated reoriented coding regions of *Aeropyrum pernix K* genome [15] (NC000854 [12]) versus all genomes.

The unwrapped phase, which is linked by (3) to the nucleotide positive and negative transition frequencies, shows little or no change when replacing the chromosome nucleotide sequence with the concatenated sequence of reoriented coding regions. As explained, the reorientation of the inverse coding regions consists in their reversal and switching of their strands.

The model also explains the finding that the unwrapped phase, which reveals second-order statistical features, has an almost linear variation even for eukaryote chromosomes [1, 2, 3, 4, 5, 6, 7] despite their very high fragmentation and quasirandom distribution of direct and inverse ORFs, while the cumulated phase, linked to the frequency of nucleotides along the DNA strands, displays only a slight drift close to zero. Figure 6 gives the cumulated phase and the unwrapped phase along the complete chromosome 4 [14] of *Mus musculus* (accession NT019246 [12]). The unwrapped phase increases almost linearly (actually there are two domains of quasilinearity with distinct slopes), while the cumulated phase remains almost zero (at the scale of the plot). Similar results have been obtained for all *Mus musculus* and *Homo sapiens* chromosomes.

The reversal of all inverse segments along the same positive direction, as performed for prokaryotes, would most probably reveal a similar "hidden linear variation" of the cumulated phase. Unfortunately, for eukaryotes, the information about the OFR orientation is not sufficient to perform the reordering, because the extension of the coding regions is only a small fraction from the total length of the chromosome. We illustrate the way the "hidden" linear variation of the cumulated phase could be revealed by DNA segment reorientation, by using again the case of a prokaryote, the aerobic hyperthermophilic crenarchaeon *Aeropyrum pernix K*, for which the genome has been completely sequenced [12, 14]. Figure 7 presents the cumulated and the unwrapped phases of the genomic signal for the entire genome comprising 1669695 base pairs. The unwrapped phase varies almost linearly, like in all the other investigated prokaryote and eukaryote genomes [1, 2, 3, 4, 5, 6, 7], confirming the rule stated in Section 1 and explained in this paper. The cumulated phase decreases irregularly, an untypical behavior for prokaryotes that tend to have a regular piecewise linear variation of the cumulated phase, as shown above. Figure 7 also shows the cumulated and unwrapped phases of the signal that correspond to a sequence obtained by concatenating the 1839 coding regions in the genome after reorienting them all in the same reference direction. The new sequence comprises only the 1553043 base pairs involved in the coding regions for which the sense information is available; the intergenic regions, for which this information is missing, have been left out. As seen in the figure, the cumulated phase changes to a uniform, almost linear, increase while the unwrapped phase remains practically unchanged.

5. CONCLUSION

DNA sequences of complete chromosomes or sequences obtained by concatenating all reoriented coding regions of chromosomes have been converted into genomic signals by using a nucleotide complex representation derived from the nucleotide tetrahedral representation. Some large-scale features of the resulting genomic signals have been analyzed. The cumulated phase and unwrapped phase of genomic signals are correlated with the statistical distribution of bases and base pairs, respectively. The paper presents a model of the longitudinal structure of the chromosomes that explains the almost linear variation of the unwrapped phase of the complex genomic signals for all prokaryotes and eukaryotes [1, 2, 3, 4, 5, 6, 7]. The linearity of the cumulated phase for the reordered ORFs, reflecting a large-scale homogeny of the nucleotide distribution in such sequences, on one hand, and the taxon specific variation of the cumulated phase for the actual nucleic DNA strands, on the other, suggest the hypotheses of a primary ancestral genomic material and of a functional role of the particular orientation of direct and inverse DNA segments that generate specific densities of the first- and second-order repartition of nucleotides along chromosomes. The relevance of these large-scale features of chromosomes in the control of the crossing-over/recombination process, the identification of the interacting regions of chromosomes, and the separation of species, as well as the mechanisms that generate the specific arrangements of direct and inverse ORFs remain to be further investigated.

REFERENCES

[1] P. Cristea, "Genomic signals for whole chromosomes," in *Manipulation and Analysis of Biomolecules, Cells, and Tissues*, vol. 4962 of *Proceedings of SPIE*, pp. 194–205, San Jose, Calif, USA, January 2003.

[2] P. Cristea, "Large scale features in DNA genomic signals," *Signal Processing*, vol. 83, no. 4, pp. 871–888, 2003.

[3] P. Cristea, "Conversion of nucleotides sequences into genomic signals," *J. Cell. Mol. Med.*, vol. 6, no. 2, pp. 279–303, 2002.

[4] P. Cristea, "Genetic signal representation and analysis," in *Functional Monitoring and Drug-Tissue Interaction*, vol. 4623 of *Proceedings of SPIE*, pp. 77–84, San Jose, Calif, USA, January 2002.
[5] P. Cristea, "Genetic signal analysis," in *Proc. 6th International Symposium on Signal Processing and Its Applications (ISSPA '01)*, pp. 703–706, Kuala Lumpur, Malaysia, August 2001.
[6] P. Cristea, "Genetic signals," *Rev. Roum. Sci. Techn. Electrotechn. et Energ.*, vol. 46, no. 2, pp. 189–203, 2001.
[7] P. Cristea and R. Tuduce, "Signal processing of genomic information: Mitochondrial genomic signals of hominidae," in *Proc. 4th EURASIP Conference Focused on Video/Image Processing and Multimedia Communications (EC-VIP-MC '03)*, Zagreb, Croatia, July 2003.
[8] E. Chargaff, "Structure and function of nucleic acids as cell constituents," *Federation Proceeding*, vol. 10, pp. 654–659, 1951.
[9] J. M. Freeman, T. N. Plasterer, T. F. Smith, and S. C. Mohr, "Patterns of genome organization in bacteria," *Science*, vol. 279, no. 5358, pp. 1827–1832, 1998.
[10] A. Grigoriev, "Analyzing genomes with cumulative skew diagrams," *Nucleic Acids Research*, vol. 26, no. 10, pp. 2286–2290, 1998.
[11] J. R. Lobry, "Asymmetric substitution patterns in the two DNA strands of bacteria," *Molecular Biology and Evolution*, vol. 13, no. 5, pp. 660–665, 1996.
[12] National Center for Biotechnology Information, National Institutes of Health, National Library of Medicine, GenBank, http:// www.ncbi.nlm.nih.gov/genoms/.
[13] J. Parkhill, G. Dougan, K. D. James, et al., "Complete genome sequence of a multiple drug resistant Salmonella enterica serovar Typhi CT18," *Nature*, vol. 413, no. 6858, pp. 848–852, 2001.
[14] J. Kawai, A. Shinagawa, K. Shibata, et al., "Functional annotation of a full-length mouse cDNA collection," *Nature*, vol. 409, no. 6821, pp. 685–690, 2001, RIKEN Genome Exploration Research Group Phase II Team and the FANTOM Consortium.
[15] Y. Kawarabayasi, Y. Hino, H. Horikawa, et al., "Complete genome sequence of an aerobic hyper-thermophilic crenarchaeon, Aeropyrum pernix K1," *Journal of DNA Research*, vol. 6, no. 2, pp. 83–101, 1999.

Paul Dan Cristea graduated from the Faculty of Electronics and Telecommunications, Politehnica University of Bucharest (PUB) in 1962, and the Faculty of Physics, PUB, as head of the series. He obtained the Ph.D. degree in technical physics from PUB, in 1970. His research and teaching activities have been in the fields of genomic signals, digital signal and image processing, connectionist and evolutionary systems, intelligent e-learning environments, computerized medical equipment, and special electrical batteries. He is the author or coauthor of more than 125 published papers, 12 patents, and contributed to more than 20 books in these fields. Currently, he is the General Director of the Biomedical Engineering Center of PUB and Director of the Romanian Bioinformatics Society.

EURASIP Journal on Applied Signal Processing 2004:1, 138–145

A Genetic Programming Method for the Identification of Signal Peptides and Prediction of Their Cleavage Sites

David Lennartsson
Saida Medical AB, Stena Center 1A, SE-412 92 Göteborg, Sweden
Email: david.lennartsson@saida-med.com

Peter Nordin
Department of Physical Resource Theory, Chalmers University of Technology, SE-412 96 Göteborg, Sweden
Email: peter.nordin@mc2.chalmers.se

Received 28 February 2003; Revised 31 July 2003

A novel approach to signal peptide identification is presented. We use an evolutionary algorithm for automatic evolution of classification programs, so-called programmatic motifs. The variant of evolutionary algorithm used is called genetic programming where a population of solution candidates in the form of full computer programs is evolved, based on training examples consisting of signal peptide sequences. The method is compared with a previous work using artificial neural network (ANN) approaches. Some advantages compared to ANNs are noted. The programmatic motif can perform computational tasks beyond that of feed-forward neural networks and has also other advantages such as readability. The best motif evolved was analyzed and shown to detect the h-region of the signal peptide. A powerful parallel computer cluster was used for the experiment.

Keywords and phrases: signal peptides, genetic programming, bioinformatics, programmatic motif, artificial neural networks, cleavage site.

1. INTRODUCTION

The huge and growing amount of unanalyzed data present in genetic research creates a demand for automatic methods for classification of proteins and protein properties. Automatic mechanical means for property screening of interesting proteins would accelerate the process of finding new drug candidates.

Classification rules for the processing of amino acid sequences can be obtained either by human design or by a mechanical process, the latter often through the use of machine-learning algorithms.

A signal peptide is a short region of amino acid residues situated at the N-terminal part of some peptide chains. Commonly, signal peptides are referred to as the address tags within the cell since they control the transport of proteins through the *secretory pathway*, the mechanism that moves proteins through cell membranes. These proteins are produced by ribosomes in the cytoplasm but the produced peptide does not fold to become a protein at this stage. Instead, the first part of the peptide, the signal peptide, attaches itself to a *translocon* in the membrane. This binding opens a channel and the peptide starts to transport itself through the translocon channel. After transportation through the membrane, the signal peptide cleaves from the protein's peptide and the channel is closed. The protein's peptide is now free and can fold itself to become an active, or *mature*, protein.

The existence of a signaling mechanism in the cell was first postulated by Günther Blobel in 1971. After a series of experiments, he came to the correct conclusion that the signal, or address tag, was coded with amino acids as part of the peptide and the transport went through channels in the membranes. Later, Blobel could verify that the process was universal. The same mechanisms work not only in animal cells but also in bacteria, yeast, and plants. For his work, Blobel received the Nobel prize in medicine in 1999.

The knowledge about signal peptides has been instrumental in understanding some hereditary diseases caused by proteins not reaching their intended destination. It is also believed that signal peptides will help in engineering yeast cells into drug factories. Drugs could then be delivered from the cells through secretion.

2. PREVIOUS RESEARCH

An early approach to signal peptide classification is the matrix method used by von Heijne in [1]. The matrix was

constructed out of the known signal peptides at the time and gave results of a sequence level performance of 78% correct classification for eukaryotic sequences.

Nielsen et al. [2] improved on the weight matrix method and carried out an experiment where they used feed-forward artificial neural networks trained with backpropagation to predict if a peptide had a signal peptide attached or not.

To compare this method with the more traditional weight matrix method, they started with a recalculation of the matrix weights using the sequences already known. In 1996, the number of known signal peptides was 5–10 times greater than in 1986. However, the results were considerably worse than the results obtained by von Heijne in 1986, and only 66% of the eukaryotic sequences were classified successfully. Nielsen et al. attributes the failure either to larger variation in the signal peptides found since 1986 or to more frequent errors in the dataset. The 1986 dataset was hand-compiled while Nielsen et al. used an automatic method.

The neural network method combined the results of two individually trained networks that were trained on different tasks. The first network tried to predict if a specific position in the sequence was part of the signal peptide or not while the second network tried to predict if the position was the cleavage site. The combined output from the two networks was based on changes in the output from the first network close to peaks in the output from the second network. Together, the two networks managed to predict 70% of the eukaryotic sequences correctly and 68% of the sequences from the human dataset. Their method and signal peptide identification service is known as *signalP*.

The use of genetic programming (GP) for protein classification tasks has been pioneered by Koza. In [3], he uses it to find protein *motifs* and in [4] he coined the term *programmatic motif* and used the method for evolving a rule that predicted the cellular location of a given protein. Both experiments produced results better than any other method at the time, including hand-crafted motifs.

3. DATA

In our experiments, we used the data Nielsen et al. made public on their ftp-server [5]. It is the same data they used in their own experiments and the data originates from SWISS-PROT version 29 [6]. Nielsen et al. started with selecting sequences marked with SIGNAL. From the SIGNAL group, they removed all proteins where they could suspect that they had been tagged as SIGNAL in a nonverified way, that is, by the use of prediction algorithms or guessing. As a background, they chose different known cytoplasmic and nuclear proteins. Here they also removed all entries that seemed to be nonverified.

Furthermore, they also compared the data and excluded sequences that were too similar to others. In this way redundancy in the dataset was reduced. For a more detailed description of the extraction and preparation of the dataset, see [2, 7].

Nielsen et al. performed their experiment on several different groups of proteins including human, *E. coli*, eukaryotes, and gram+ and gram− bacteria, with similar results for all groups. For experiments described in this paper, we chose to work only with the human dataset.

In our experiments, the data was split into two sets: one *training set* consisting of 176 background proteins and 291 signal peptides and one *validation set* consisting of 75 background proteins and 125 signal peptides. For every position in the peptide sequence, the dataset included information telling whether it was part of a mature protein or part of a signal peptide. An excerpt from the dataset is shown in Figure 1.

The peptide sequences were truncated after 70 amino acids for background proteins. In the case of signal peptides, the signal part and the first 30 positions of the mature protein were kept. This makes sense since the process of translocation starts before the whole peptide is produced by the ribosome.

4. METHOD

We have used the machine-learning technique GP. GP is a branch of evolutionary algorithms where computer programs are evolved from first principles to solve a problem specified by a fitness function. Although GP has many features in common with other branches of evolutionary computation, such as genetic algorithms (where often fixed-length binary genomes are evolved), the solutions evolved by a GP system are more complex and can solve harder problems; they are often complete *programs* or *algorithms*.

In GP, a population of *solution candidates*, individual programs, is kept and these individuals compete for the right to reproduce. During mating, variations are introduced in the offspring's genome by the use of *genetic operators*. Two common simulated operators are mutation and sexual recombination. The undirected mechanisms of random variation combined with selection through survival of the fittest leads to evolution. The competing individuals in the population will usually improve over time at the task by which they are graded, and the more *fit* individuals survive and proliferate.

The solution candidates, or the individuals, have two appearances, the *genotype* and the *phenotype*. The genotype is the genome, the recipe that builds the phenotype, and the behavior of the program. In GP, the phenotype is a program being executed on a real or simulated *machine*. Depending on the phenotype's performance, the genotype may reproduce. Since the selection criterion is defined as an external property, the algorithm might be seen as more similar to breeding than to actual evolution.

Three different types of genomes are common in GP: tree-like, linear, and graph-like. In this experiment, a linear representation of the genome was used. For more background on GP and discussions about genome, representation, theory, and different selection mechanisms, see [8, 9, 10, 11].

The individuals in the population had variable-length genomes that could contain up to 300 instructions. Evolution started with a population with genomes of random length and random content (genes).

```
 0
70 RPB2_HUMAN    DNA-DIRECTED RNA POLYMERASE II 140 KD POLYPEPTIDE
MYDADEDMQYDEDDDEITPDLWQEACWIVISSYFDEKGLVRQQLDSFDEFIQMSVQRIVEDAPPIDLQAE
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM


 1
51 1OKS_HUMAN     21 CLARA CELLS 10 KD SECRETORY PROTEIN PRECURSOR (CC10).
MKLAVTLTLVTLALCCSSASAEICPSFQRVIETLLMDTPSSYEAAMELFSP
SSSSSSSSSSSSSSSSSSSSCMMMMMMMMMMMMMMMMMMMMMMMMMMMM
```

Figure 1: All the sequences have a class, a name, and a specification of which kind of peptide the acid is part of. Here, S means that the amino acid is part of the signal peptide while C and M are parts of the mature protein; C marks the cleavage site.

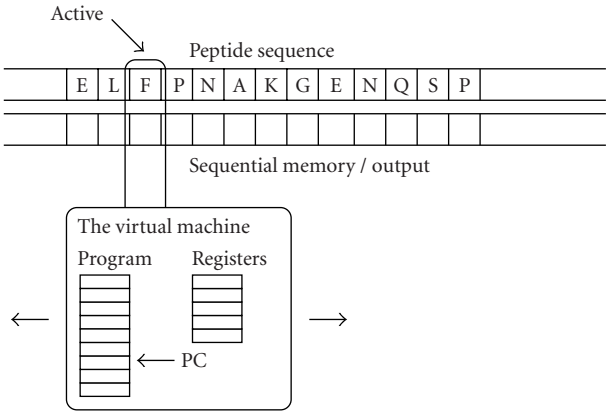


Figure 2: The evolved program instructs the virtual machine to move along the sequence and to perform calculations on registers and writing to memory.

4.1. The virtual machine

The linear genomes of the individuals are interpreted as a computer program by a *virtual machine*. The virtual machine used was implemented as a register machine. The machine had the ability to analyze the peptide sequence, perform arithmetics with five registers, and use a sequential memory. A schematic of the machine is shown in Figure 2.

Each position in the individual's genome represents a complete instruction and is encoded as a 32-bit integer. The first eight bits encodes the operation while the following three bytes are passed as arguments. The most common argument is a pointer to a register, but depending on the operation, it could also be interpreted as a real-valued constant or a relative program address. Regardless of how a gene is coded, it is always reinterpreted as a valid instruction with valid arguments.

The following operations were supported by the machine:

(i) Boolean operators: and, or, xor, not;
(ii) register setting operators: one, clear, set;
(iii) arithmetic operators: add, sub, mul, div, sigmoid;
(iv) branching operators: ifgtz, jmp, jmpgtz;
(v) head-moving operators: for, rev, home;
(vi) memory-altering operators: read, write;
(vii) amino acid residue detecting operators: ala, arg, asn, asp, cys, glu, gln, gly, his, ile, leu, lys, met, phe, pro, ser, thr, trp, tyr, val, aliphatic, aromatic, charged, hydrophobic, negative, polar, positive, small, tiny.

The application-specific operators in this virtual machine are the amino acid residue detecting operators. These instructions return positive if the machine is positioned over the respective target. Otherwise, a negative result is returned. There are also instructions to determine if a target has a specific chemical property.

The genome of an individual contains up to 300 instructions forming a program. The program is the individual and from this point that is what we refer to when using the word program. The virtual machine and the computational methods around it, such as fitness measurement, are referred to as the system.

The evaluation of an individual program was executed once for every peptide in the training set of fitness cases. Before every run, both registers and sequential memory were being reset to zero and the program counter was initiated to zero. The head of the virtual machine was moved to the first position in the sequence of the peptide to examine.

When the program was executed, it could instruct the virtual machine to move along the peptide chain and check for amino acid residues or properties of the residues. In between those operations, it could perform calculations on its registers and/or write to sequential memory. The sequential memory would also be treated as the output of the program. If a memory cell in the sequential memory held a value greater than zero at program termination, that cell's position was considered to be a prediction of a cleavage site. The value zero or less was considered as no prediction.

Programs terminated when reaching the end of the program or when a jump instruction instructed the machine to jump outside the program. If a program used all of its allowed executions, all branching operators were treated as NOPs (no operation) and the program terminated when the end of the program was reached. The execution limit was set to 800 instructions per run. The program would also terminate if the head was moved outside the peptide sequence.

For a more thorough description of register machine GP, see [8].

4.2. Fitness measurement

After the evaluation of the peptide sequences, the result had to be analyzed in order to assign a fitness to the individual. This process may be the most important in GP due to the principle "what you train is what you get."

The main part of the fitness was made up of errors associated with the distance between the real and the predicted cleavage site. For every predicted position, the error d^2 was added to the fitness. If the program tagged several positions, it would receive multiple penalties and thus such behavior would result in poor fitness. If no position was tagged on a signal peptide, the program would get a penalty that corresponds to a distance d of 17. The same was true for nonsignal peptides that were falsely classified to have a cleavage site.

To further guide the evolution, the fitness assigning function was made more smooth by adding a small error for every position in the memory. The system expected the program to return one for cleavage sites and minus one for every other position. Deviations from these values and an extra penalty $p = 0.15$ for falsely classified positions were added to the fitness.

Later when the system activated *parsimony pressure*, it also added a small cost associated with execution of instructions to the fitness. This cost was small enough not to affect the results of the comparison other than when the system had to choose between two equally performing individuals with different sizes. Finally, there were some penalties needed to avoid cheating and control the behavior of the program. These penalties were large. First, if a program used recursion and did not terminate before using its available 800 instructions, it would be punished for loop violation. Second, if a program produced constant output for different peptides in the set, the program would get punished.

The last punishment was received if the program tried to move the head of the virtual machine outside the peptide sequence. This was needed to avoid cheating where the program otherwise could locate the end of the sequence and count a certain number of steps back from that point. Such "cheating" solutions were often evolved by the system if no penalty was given. The total fitness function is

$$f = \frac{1}{\text{peptides}} \sum_{\text{Peptides}} \left(d^2 + \text{parsimony} + \frac{1}{\text{length}} \sum_{\text{Positions}} (e^2 + p) \right) + \text{loop\_violation} + \text{constant\_output} + \text{illegal\_move}. \quad (1)$$

The fitness was balanced in such a way that individuals first prioritize minimizing d, then e, and lastly the size of solution (parsimony pressure). The penalties for illegal behavior dominate over all of the above.

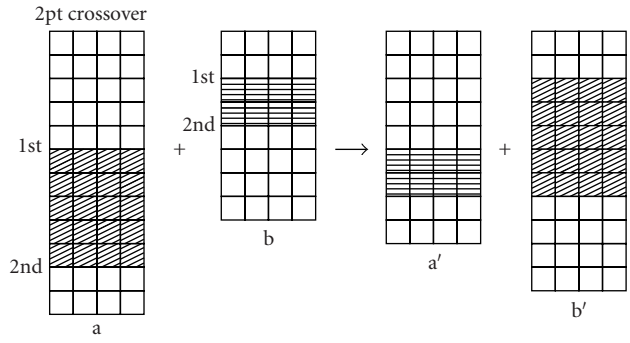


Figure 3: If sexual recombination takes place, the children (a′) and (b′) will be a combination of the parents (a) and (b) genomes. Recombination works by letting the crossover operator exchange two random parts of the genomes.

4.3. Selection and genetic operators

We used steady-state tournament selection. For every evolutionary step, four arbitrary individuals are selected. They compete against each other in two pairs and the best two individuals from the two (semifinal) games mate.

Mating produces two offspring. It can be either two perfect copies of the parents or recombinations of the parents genomes. Two-point crossover was used for recombination, shown in Figure 3. There is also a small chance that the genome of a child will be mutated at a single position.

The two less-performing individuals who were defeated in the tournament are removed while the parents and the offspring stay in the population. The process of tournaments is iterated over many generations.

4.4. Parallelization

To speed execution up, six workstations were clustered together using demes. Equal-sized subpopulations were kept in each deme and one percent of the population migrated to another deme every generation. The demes were connected with a ring-like topology.

The clustering gave a full linear speedup and there was no performance degradation due to clustering. Indications of superlinear speedups [10] were found but the experiment did not run sufficient number of times to statistically support such claims. A comparison of the evolutionary progress for a single population and a population spread over demes can be seen in Figure 4. When the system utilizes demes, the population evolves faster. It can be noted that the effort in Figure 4 is measured in computer time and that the system taking advantage of clustering was more than six times faster in real time than the system utilizing a single workstation.

5. RESULTS

The results presented in the following sections show the best performing individual. During the run, a population of twenty thousand programs was evolved for four million tournaments. Approximately eight million different solutions were tried. Parsimony pressure was added after two million

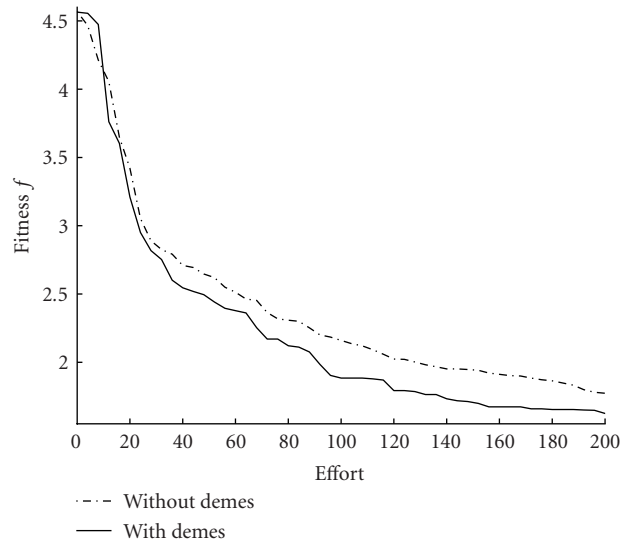


Figure 4: A comparison between a demes population and a non-demes population. The progress of evolution as the function of total computational effort. The mean fitness out of three runs plotted for both having the population spread out over demes or keeping all individuals in a single population.

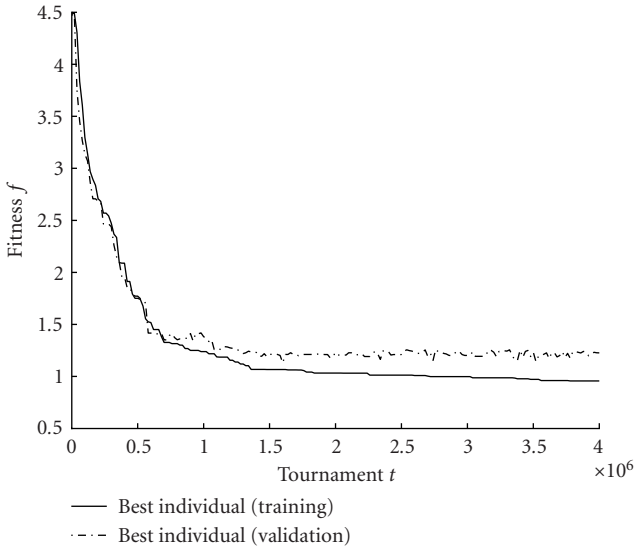


Figure 5: Fitness for population. The fitness of the two best performing individuals on training and validation data.

tournaments. During mating, there were a 98% probability of sexual recombination and 15% probability of mutation.

The best performing individual was 273 instructions long and had formed through 383 genetic operations. The whole run took about three days on standard PC hardware running at 500 MHz.

In Figure 5, we can see how the population becomes more fit over generations. Even though the best individual continues to improve on training, we do not see evidence of any overlearning. The individuals are general solutions to the problem, and fitness on validation data remains similar to that of the training fitness.

Table 1: Performance for the identification of signal peptides (best individual).

	Training set	Validation set	Whole set
Correctly identified (%)	92.5	92.5	92.5
MCC	0.84	0.84	0.84

5.1. Identification of signal peptides

The first quality measurement of the individual is how reliable the program is classifying a sequence as a signal peptide or not. Any sequence that produces an output above zero in any cell of the sequential memory is considered to be a signal peptide, while the sequences where all outputs are at or below zero are considered to be classified as background data.

We use the Matthew correlation coefficient [12] to determine the performance of a rule in addition to percentage of correctly classified signal peptides. The coefficient is defined as

$$C_{MCC} = \frac{N_{tp}N_{tn} - N_{fp}N_{fn}}{\sqrt{(N_{tn} + N_{fn})(N_{tn} + N_{fp})(N_{tp} + N_{fn})(N_{tp} + N_{fp})}}. \quad (2)$$

The coefficient C_{MCC} equals one for a perfect prediction, minus one for a total opposite prediction, and zero for a completely random prediction. The variables N_{tp}, N_{tn}, N_{fp}, and N_{fn} represent the number of correctly classified positives, correctly classified negatives, falsely classified positives, and falsely classified negatives, respectively.

The performance of the best individual on the task of identifying signal peptides is presented in Table 1. The individual managed equally well on the training and validation cases and actually had a lower fitness on the validation data than on the training set which indicates that there was no overtraining.

5.2. Predicting cleavage site location

After identifying which sequences that include a signal peptide, we would like to know where their cleavage sites are located. The individuals are trained to minimize the distance between predicted and actual cleavage site. This is introduced in the fitness as a sum over d^2.

To verify how well the individuals perform on locating the cleavage site, the percentage of signal peptide sequences with correctly predicted cleavage sites was measured. In this case, a correct prediction is a predicted cleavage site at most two positions away from the real site.

The results of the same best individual as in the previous sections are presented in Table 2. To further know if this result was better than a random guess, the average distance between the predicted cleavage site and the real cleavage site was calculated.

Table 2: Performance for the prediction of cleavage sites (best individual).

	Training set	Validation set	Whole set
Correctly predicted (%)	53.3	61.6	55.8
Mean d^2	12.2	12.7	12.3

To put the measured distance d^2 into perspective, a couple of different test measurements were carried out. First we measured how large the mean value of d^2 would be if the prediction algorithm chose random points distributed uniformly between the two extreme positions for cleavage sites found in the whole dataset. The mean, out of a 100 test runs, yielded a d^2 of 194. This large d^2 is expected since the distribution of cleavage site positions is far from uniform. Next step was to use the discrete frequency distribution in the dataset to transform the randomness to follow the distribution. These runs gave a mean square distance of 55. Thus, no random solutions could compete with the measured distance of the best individual.

Earlier in the studies, the system had produced individuals with constant output which managed to reach quite low fitness and therefore the mean distance for various constant solutions is needed to be measured. The best constant solution was the one stating that the cleavage site was positioned at position 24 in the peptide sequence. This solution had a mean d^2 of 28.

In comparison with the tests above, it is clear that the best individual evolved far from being a random guess or optimal constant solution.

5.3. Analysis of the best individual program

One of the often stated advantages of GP compared, for instance, to artificial neural networks is the ability to produce the result in a human readable form. It is much harder to analyze the weights and get a grip of how an artificial neural network is calculating its results than to analyze program code.

In our case, the task of analysis takes some effort since we let the program evolve without any constraints on its architecture. The individuals could evolve loops and subfunctions with the help of branching instructions. Since the individuals only had one single linear genome, these functions sometimes overlapped. A loop may partially overlap with another loop and some parts of the code will be used differently at different times. Still the function of an individual is not that hard to understand.

Although the mechanism for targeting signal peptides work similar in all organisms, the signal peptides do not share one common sequence. They do however share a common structure. There are some simple rules of thumb to detect a signal peptide. First the sequence should start with a short region, usually of positively charged amino acids, called the *n-region* at the N-terminal of the peptide. It is followed by a somewhat longer region of hydrophobic amino acids called the *h-region*. Between the hydrophobic region and the cleavage site is a short region consisting mainly of polar and uncharged amino acids named the *c-region*. At the positions before the cleavage site, a pattern called the $(-3, -1)$ rule is common. It states that position -1 and -3 relatively to the cleavage site should be occupied by small and neutral residues. The amino acid residue at position -2 can however be an aromatic, charged, or large polar residue.

A quick analysis of the program from the best individual revealed that at most 30% of the instructions contributed to the solution. The others are known in genetic programming as *introns*, genes/instructions that are inactive. Introns are also common in nature and could among other functions be a product of evolution's desire to protect important information in the genome from mutations. In GP, they consist of operations where the results produced will be overwritten by another operator without being used anywhere in between.

The evolved program consists mainly of two parts where the first part is made up of four nested loops. The program will stay inside these loops and iterate over the peptide sequence until it has come across four aliphatic residues and has not detected any proline or arginine. If encountered, the program will go back and loop some more. When this happens, the program moves around eleven positions forward. There, it performs a simple check and marks the position as a cleavage site if there is no tryptophan there. Tryptophan is a large aromatic residue. Aliphatic residues are also hydrophobic, so it seems that our program has found a simple rule relying on finding the h-region, moving across the most common number of positions and marking the cleavage site if not completely wrong. The code seems very simple but still the program can discriminate between signal peptides and other proteins with good accuracy. It has also successfully predicted cleavage sites as close to the N-terminal as 17 positions and as far away as 37 positions, so the rule spans over signal peptides with quite different characteristics.

6. COMPARISON WITH PREVIOUS METHODS

Nielsen et al. presented their results on the task of the identification of signal peptides with the help of Matthews correlation coefficient and reported it to be $C_{SP} = 0.96$, as the best, for the human dataset. This is a good value but they tried several ways of interpreting the output from the network and also optimized the threshold value used in the interpretation. When they only used their cleavage site predicting network, which is more similar to the approach presented in this paper, and used the highest output to determine if a sequence has a signal peptide or not, they got a $C_{SP} = 0.71$ which is worse than the $C_{SP} = 0.84$ reached in this experiment.

When it comes to predicting the cleavage site, Nielsen et al. reported a 68.0% success rate on the human dataset using the combined output from two different neural networks. The weight matrix method with newly calculated weights scored 66.7%. According to a survey performed by Emanuelsson et al. [13], *TargetP*, the successor to signalP,

correctly predicted 81.1% of the cleavage sites within two positions from the real site. The best individual in our experiment scored 55.8%.

Although this is comparing apples to oranges, it can be interesting to note how much parameters are included in the solutions. The two networks used to classify human signal peptides contained in total 3080 real-valued parameters while the program produced through GP had a length of 273 32-bit instructions. About 30% of these instructions were actually used in the solution. The instruction set is highly redundant and could easily fit into a 16-bit representation. The evolved program can be described using much less information than the neural network.

GP is also generally less sensitive to initial parameter settings than neural networks, making it possibly a more robust search tool.

Another difference between the systems is the ability to learn from the solution derived from the method. The resulting program from the GP system is available in a human-readable form, although it may take some work to sort it out. This way, the GP approach holds promise for the future since it is not only a program that predicts, but also it can produce new human knowledge.

7. DISCUSSION

The evolved programs have a quite complex architecture with the ability to create iterations and conditional loops. The programs evolved by GP can therefore express completely different patterns than practically possible with artificial neural networks. This may also make a hybrid method between neural networks and a candidate for future research.

A great deal of effort was spent to prevent programs from "cheating." Examples of cheating would be to count positions from the end of the peptide in the dataset. Although it is clear that the predictive performance of the neural networks is not affected by this kind of cheating, it is not fully evident from publications if enough effort is spent on preventing the network from building up the kind of function needed for all kinds of possible cheating.

Our results are not verified with cross-validation. Instead, we have relied solely on the use of separate training and validation sets. Since no overlearning has been detected, we judge this method as sufficient. We would however like to use cross-validation in the future but there are questions regarding its accuracy in combination with evolutionary techniques.

The system identified and extracted a rule similar to a hand-discovered rule within signal peptide sequence analysis. On the task of the identification of signal peptides, the evolved rule faired well. The combined score of the neural networks was however significantly better at prediction of the cleavage sites.

The interpretability of solutions enables the GP technique to be used for extraction of new knowledge regarding cleavage sites and signal peptides. The clear text output enables reformulation as human knowledge.

8. CONCLUSION

We have shown that GP can be used to extract features in peptide sequences. The resulting "programmatic motifs" have a high expressiveness and can express other information than practically possible with, for example, neural networks.

Unlike many other methods, the resulting program is available in a human-readable form and is interpretable. An analysis of the program showed that it has evolved a rule that relied heavily on finding the hydrophobic core in the signal peptide.

GP is still a young research field and this report describes one of the first experiments on peptide classification with this method. Our results points to the feasibility of further use of genetic programming in sequence analysis tasks.

ACKNOWLEDGMENT

Peter Nordin gratefully acknowledges the support from Owe Orwar.

REFERENCES

[1] G. von Heijne, "A new method for predicting signal sequence cleavage sites," *Nucleic Acids Res.*, vol. 14, no. 11, pp. 4683–4690, 1986.

[2] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne, "A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites," *Int. J. Neural Syst.*, vol. 8, no. 5-6, pp. 581–599, 1997.

[3] J. R. Koza and D. Andre, "Automatic discovery of protein motifs using genetic programming," in *Evolutionary Computation: Theory and Applications*, X. Yao, Ed., World Scientific, Singapore, 1996.

[4] J. R. Koza, F. Bennett, and D. Andre, "Using programmatic motifs and genetic programming to classify protein sequences as to extracellular and membrane cellular location," in *Evolutionary Programming VII: Proceedings of the 7th Annual Conference on Evolutionary Programming*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., vol. 1447, Springer-Verlag, San Diego, Calif, 1998.

[5] H. Nielsen, S. Brunak, J. Engelbrecht, and G. von Heijne, *Data from signalP ftp-site*, http://www.cbs.dtu.dk/ftp/signalp/.

[6] A. Bairoch and B. Boeckmann, "The SWISS-PROT protein sequence data bank: current status," *Nucleic Acids Res.*, vol. 22, no. 17, pp. 3578–3580, 1994.

[7] H. Nielsen, J. Engelbrecht, G. von Heijne, and S. Brunak, "Defining a similarity threshold for a functional protein sequence pattern: the signal peptide cleavage site," *Proteins*, vol. 24, pp. 165–177, 1996.

[8] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming: An Introduction*, Morgan Kaufmann, San Francisco, Calif, 1998.

[9] J. R. Koza, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, Mass, 1992.

[10] J. R. Koza, F. H. Bennett III, D. Andre, and M. A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann, San Francisco, Calif, 1999.

[11] R. Poli and W. B. Langdon, *Foundations of Genetic Programming*, Springer-Verlag, Berlin, 2002.

[12] B. W. Matthews, "Comparison of predicted and observed secondary structure of T4 phage lysozyme," *Biochemica et Biophysica Acta.*, vol. 405, no. 2, pp. 442–451, 1975.
[13] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne, "Predicting subcellular localization of proteins based on their N-terminal amino acid sequence," *J. Molecular Biology*, vol. 300, no. 4, pp. 1005–1016, 2000.

David Lennartsson has been working as a Consultant in software development for several years. He received his M.S. degree in engineering physics from Chalmers University of Technology, Sweden, in 2003. This paper is originally based on his thesis work. Currently, he is focusing his research efforts on systems for knowledge extraction and decision support using intelligent heuristics such as genetic programming. Mr. Lennartsson is one of the founders of SAIDA Medical which develops methods for automatic statistical inference and modelling.

Peter Nordin received his M.S. degree in computer science and engineering from Chalmers University of Technology, Sweden, in 1989, and his Ph.D. degree in computer science from the University of Dortmund, Germany, in 1997. He has worked for several years as a Researcher and Consultant in the area of knowledge-based systems, artificial intelligence, and evolutionary algorithms at Infologics AB, a subsidiary of Swedish telecom. Dr. Nordin is a Cofounder of Dacapo AB, a Swedish consulting and research company specialised in the state-of-the-art information technology, and an Inventor of the patented AIM-GP genetic programming method, a very efficient approach to GP. He has published 90 papers on genetic programming. He has been Program Cochair of EuroGP'99, Second European Workshop on Genetic Programming, and is in the editorial board of the Journal of Genetic Programming and Evolvable Hardware. Dr. Nordin has been a member of several European research projects. Since 1998, he has been an Associate Professor in the Complex Systems Group at Chalmers University of Technology.

EURASIP Journal on Applied Signal Processing 2004:1, 146–153

Genomic Signal Processing: The Salient Issues

Edward R. Dougherty
Department of Electrical Engineering, Texas A&M University, 3128 TAMU College Station, TX 77843-3128, USA
Email: e-dougherty@tamu.edu

Ilya Shmulevich
Department of Pathology, The University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA
Email: is@ieee.org

Michael L. Bittner
Molecular Diagnostics and Target Validation Division, Translational Genomics Research Institute, Tempe, AZ 85281, USA
Email: mbittner@tgen.org

Received 10 October 2003

This paper considers key issues in the emerging field of genomic signal processing and its relationship to functional genomics. It focuses on some of the biological mechanisms driving the development of genomic signal processing, in addition to their manifestation in gene-expression-based classification and genetic network modeling. Certain problems are inherent. For instance, small-sample error estimation, variable selection, and model complexity are important issues for both phenotype classification and expression prediction used in network inference. A long-term goal is to develop intervention strategies to drive network behavior, which is briefly discussed. It is hoped that this nontechnical paper demonstrates that the field of signal processing has the potential to impact and help drive genomics research.

Keywords and phrases: functional genomics, gene network, genomics, genomic signal processing, microarray.

1. INTRODUCTION

Sequences and clones for over a million expressed sequence tagged sites (ESTs) are currently publicly available. Only a minority of these identified clusters contains genes associated with a known functionality. One way of gaining insight into a gene's role in cellular activity is to study its expression pattern in a variety of circumstances and contexts, as it responds to its environment and to the action of other genes. Recent methods facilitate large-scale surveys of gene expression in which transcript levels can be determined for thousands of genes simultaneously. In particular, expression microarrays result from a complex biochemical-optical system incorporating robotic spotting and computer image formation and analysis. Since transcription control is accomplished by a method that interprets a variety of inputs, we require analytical tools for expression profile data that can detect the types of multivariate influences on decision making produced by complex genetic networks. Put more generally, signals generated by the genome must be processed to characterize their regulatory effects and their relationship to changes at both the genotypic and phenotypic levels. Two salient goals of functional genomics are to screen for key genes and gene combinations that explain specific cellular phenotypes (e.g., disease) on a mechanistic level, and to use genomic signals to classify disease on a molecular level.

Genomic signal processing (GSP) is the engineering discipline that studies the processing of genomic signals. Owing to the major role played in genomics by transcriptional signaling and the related pathway modeling, it is only natural that the theory of signal processing should be utilized in both structural and functional understanding. The aim of GSP is to integrate the theory and methods of signal processing with the global understanding of functional genomics, with special emphasis on genomic regulation. Hence, GSP encompasses various methodologies concerning expression profiles: detection, prediction, classification, control, and statistical and dynamical modeling of gene networks. GSP is a fundamental discipline that brings to genomics the structural model-based analysis and synthesis that form the basis of mathematically rigorous engineering.

Application is generally directed towards tissue classification and the discovery of signaling pathways, both based on the expressed macromolecule phenotype of the cell. Accomplishment of these aims requires a host of signal processing approaches. These include signal representation relevant to transcription, such as wavelet decomposition and more general decompositions of stochastic time series, and system

modeling using nonlinear dynamical systems. The kind of correlation-based analysis commonly used for understanding pairwise relations between genes or cellular effects cannot capture the complex network of nonlinear information processing based upon multivariate inputs from inside and outside the genome. Regulatory models require the kind of nonlinear dynamics studied in signal processing and control, and in particular the use of stochastic dataflow networks common to distributed computer systems with stochastic inputs. This is not to say that existing model systems suffice. Genomics requires its own model systems, not simply straightforward adaptations of currently formulated models. New systems must capture the specific biological mechanisms of operation and distributed regulation at work within the genome. It is necessary to develop appropriate mathematical theory, including optimization, for the kinds of external controls required for therapeutic intervention as well as approximation theory to arrive at nonlinear dynamical models that are sufficiently complex to adequately represent genomic regulation for diagnosis and therapy while not being overly complex for the amounts of data experimentally feasible or for the computational limits of existing computer hardware.

2. BACKGROUND

A central focus of genomic research concerns understanding the manner in which cells execute and control the enormous number of operations required for normal function and the ways in which cellular systems fail in disease. In biological systems, decisions are reached by methods that are exceedingly parallel and extraordinarily integrated, as even a cursory examination of the wealth of controls associated with the intermediary metabolism network demonstrates. Feedback and damping are routine even for the most common activities, such as cell cycling, where it seems that most proliferative signals are also apoptosis priming signals, with the final response to these signals resulting from successful negotiation of a large number of checkpoints, which themselves involve further extensive cross checks of cellular conditions.

Traditional biochemical and genetic characterizations of genes do not facilitate rapid sifting of these possibilities to identify the genes involved in different processes or the control mechanisms employed. Of course, when methods do exist to focus genetic and biochemical characterization procedures on a smaller number of genes likely to be involved in a process, progress in finding the relevant interactions and controls can be substantial. The earliest understandings of the mechanics of cellular gene control were derived in large measure from studies of just such a case, metabolism in simple cells. In metabolism, it is possible to use biochemistry to identify stepwise modifications of the metabolic intermediates and genetic complementation tests to identify the genes responsible for catalysis of these steps, and those genes and *cis*-regulator elements involved in the control of their expression. Standard methods of characterization guided by some knowledge of the connections could thus be used to identify process components and controls. Starting from the basic outline of the process, molecular biologists and biochemists have been able to build up a very detailed view of the processes and regulatory interactions operating within the metabolic domain.

In contrast, for most cellular processes, general methods to implicate likely participants and to suggest control relationships have not emerged. The resulting inability to produce overall schemata for most cellular processes has meant that gene function is, for the largest part, determined in a piecemeal fashion. Once a gene is suspected of involvement in a particular process, research focuses on the role of that gene in a very narrow context. This typically results in the full breadth of important roles for well-known, highly characterized genes being slowly discovered. A particularly good example of this is the relatively recent appreciation that oncogenes such as Myc can stimulate apoptosis in addition to proliferation [1].

Recognition of this bottleneck has stimulated the field's appetite for methods that can provide a wider experimental perspective on how genes interact. High-throughput microarray technology, which facilitates large-scale surveys of gene expression, can now provide enormous data sets concerning transcriptional levels [2, 3, 4, 5]. As these measurements are snapshots of the types of levels of transcripts required to achieve or maintain the cell state being observed, they constitute a de facto source of information about transcript interactions involved in gene regulation.

Analysis of this data can take two routes: gene-by-gene analysis or multivariate analysis of interactions among many genes simultaneously. Correlation and other similarity measures can identify common elements of a cell's response to a particular stimulus and thus discern some groups of genes; however, correlation does not address the fundamental problem of determining the sets of genes whose actions and interactions drive the cell's decision to set the transcriptional level of a particular gene. Because transcriptional control is accomplished by a complex method that interprets a variety of inputs [1, 6, 7], the development of analytical tools that detect multivariate influences on decision-making present in complex genetic networks is essential. To carry out such an analysis, one needs appropriate analytical methodologies.

As a discipline, signal processing involves the construction of model systems. These can be composed of various mathematical structures, such as systems of differential equations, graphical networks, stochastic functional relations, and simulation models. By its nature, signal processing draws upon many related disciplines, including estimation, classification, pattern recognition, control, information, networks, computation, statistics, imaging, coding, and artificial intelligence. These in turn draw upon signal processing to the extent that their application involves processing signals.

Numerous mathematical and computational methods have been proposed for construction of formal models of genetic interactions. Many of these models have the following general characteristics:

(1) the models essentially represent *systems* in that they

(a) characterize an interacting group of components forming a whole,
(b) can be viewed as a process that results in a transformation of signals,
(c) generate outputs in response to input stimuli;

(2) the models are *dynamical* in that they
(a) capture the time-varying quality of the physical process under study,
(b) can change their own behavior over time;

(3) the models can be considered generally *nonlinear* in that the interactions within the system yield behavior more complicated than the sum of the behaviors of the agents.

The preceding characteristics are representatives of nonlinear dynamical systems. These are composed of states, input and output signals, transition operators between states, and output operators. In their most abstract form, they are very general. More mathematical structure is provided for particular application settings. For instance, in computer science they can be structured into the form of dataflow graphical networks that model asynchronous distributed computation, a model that is very close to genomic regulatory models. There have been many attempts to model gene regulatory networks including probabilistic graphical models, such as Bayesian networks [8, 9, 10, 11], neural networks [12, 13], differential equations [14], Boolean [15] and probabilistic Boolean networks [16, 17], and models including stochastic components on the molecular level [18].

As we look towards medical applications based on functional genomics, dynamical modeling is at the center. Somogyi and Greller [19] give the following areas in which dynamical modeling will play a "pivotal role":

(i) stimulus-response interactions,
(ii) prediction of new targets based on pathway context,
(iii) potential use of combinatorial therapies,
(iv) pathway responses including the understanding of reactive or compensatory behavior,
(v) stress and toxic response mechanisms,
(vi) off-target effects of therapeutic compounds,
(vii) pharmacodynamics,
(viii) characterization of disease states by dynamical behavior,
(ix) gene expression and protein expression signatures for diagnostics,
(x) design of optimized time-dependent dosing regimens.

As we consider the salient issues of GSP, it should become evident that the preceding list offers a call for a major effort on the part of the signal processing community to apply its store of knowledge to genetic science and medicine.

3. TECHNOLOGY

A cell relies on its protein components for a wide variety of its functions, including energy production, biosynthesis of component macromolecules, maintenance of cellular architecture, and the ability to act upon intra- and extra-cellular stimuli. Each cell in an organism contains the information necessary to produce the entire repertoire of proteins the organism can specify. Since a cell's specific functionality is largely determined by the genes it is expressing, it is logical that transcription, the first step in the process of converting the genetic information stored in an organism's genome into protein, would be highly regulated by the control network that coordinates and directs cellular activity. A primary means for regulating cellular activity is the control of protein production via the amounts of mRNA expressed by individual genes. The tools to build an understanding of genomic regulation of expression will involve the characterization of these expression levels. Microarray technology, both cDNA and oligonucleotide, provides a powerful analytic tool for genetic research. Since our concern in this paper is to articulate the salient issues for GSP, and not to delve deeply into microarray technology, we confine our brief discussion to cDNA microarrays.

Complementary DNA microarray technology combines robotic spotting of small amounts of individual, pure nucleic acid species on a glass surface, hybridization to this array with multiple fluorescently labeled nucleic acids, and detection and quantitation of the resulting fluor-tagged hybrids by a scanning confocal microscope. A basic application is quantitative analysis of fluorescence signals representing the relative abundance of mRNA from distinct tissue samples. Complementary DNA microarrays are prepared by printing thousands of cDNAs in an array format on glass microscope slides, which provide gene-specific hybridization targets. Distinct mRNA samples can be labeled with different fluors and then co-hybridized onto each arrayed gene. Ratios (or sometimes the direct intensity measurements) of gene expression levels between the samples can be used to detect meaningfully different expression levels between the samples for a given gene. Given an experimental design with multiple tissue samples, microarray data can be used to cluster genes based on expression profiles, to characterize and classify disease based on the expression levels of gene sets, and for other signal processing tasks.

A typical glass-substrate and fluorescent-based cDNA microarray detection system is based on a scanning confocal microscope, where two monochrome images are obtained from laser excitations at two different wavelengths. Monochrome images of the fluorescent intensity for each fluor are combined by placing each image in the appropriate color channel of an RGB image. In this composite image, one can visualize the differential expression of genes in the two cell types: test sample typically placed in red channel, and the reference sample in the green channel. Intense red fluorescence at a spot indicates a high level of expression of that gene in the test sample with little expression in the reference sample. Conversely, intense green fluorescence at a spot indicates relatively low expression of that gene in the test sample compared to the reference. When both test and reference samples express a gene at similar levels, the observed array spot is yellow. Assuming that specific DNA products from two samples have an equal probability of hybridizing to the specific target, the fluorescent intensity measurement

is a function of the amount of specific RNA available within each sample, provided that samples are well mixed and there is sufficiently abundant cDNA deposited at each target location.

When using cDNA microarrays, the signal must be extracted from the background. This requires image processing to extract signals arising from tagged reverse-transcribed cDNA hybridized to arrayed cDNA locations [20], and variability analysis and measurement quality assessment. The objective of the microarray image analysis is to extract probe intensities or ratios at each cDNA target location and then cross-link printed clone information so that biologists can easily interpret the outcomes and high-level analysis can be performed. A microarray image is first segmented into individual cDNA targets, either by manual interaction or by an automated algorithm. For each target, the surrounding background fluorescent intensity is estimated, along with the exact target location, fluorescent intensity, and expression ratio.

In a microarray experiment, there are many sources of variation. Some types of variation, such as differences of gene expressions, may be highly informative as they may be of biological origin. Other types of variation, however, may be undesirable and can confound subsequent analysis, leading to wrong conclusions. In particular, there are certain systematic sources of variation, usually due to specific features of the particular microarray technology, that should be corrected prior to further analysis. The process of removing such systematic variability is called normalization. There may be a number of reasons for normalizing microarray data. For example, there may be a systematic difference in quantities of starting RNA, resulting in one sample being consistently over-represented. There may also be differences in labeling or detection efficiencies between the fluorescent dyes (e.g., Cy3 or Cy5), again leading to systematic overexpression of one of the samples. Thus, in order to make meaningful biological comparisons, the measured intensities must be properly adjusted to counteract such systematic differences.

4. SALIENT ISSUES FOR GSP

In this section we address what we consider to be the salient issues for GSP: phenotype classification and genetic regulatory networks, which include expression prediction and network intervention and control. Other topics, including image processing, signal extraction, data normalization, quantization, compression, expression-based clustering, and signal processing methods for sequence analysis play necessary and supportive roles.

4.1. Classification

An expression-based classifier provides a list of genes whose product abundance is indicative of important differences in cell state, such as healthy or diseased, or one particular type of cancer or another. Among such informative genes are those whose products play a role in the initiation, progression, or maintenance of the disease. Two central goals of molecular analysis of disease are to use such information to directly diagnose the presence or type of disease and to produce therapies based on the disruption or correction of the aberrant function of gene products whose activities are central to the pathology of a disease. Correction would be accomplished either by the use of drugs already known to act on these gene products or by developing new drugs targeting these gene products.

Achieving these goals requires designing a classifier that takes a vector of gene expression levels as input and outputs a class label that predicts the class containing the input vector. Classification can be between different kinds of cancer, different stages of tumor development, or many other such differences. Classifiers are designed from a sample of expression vectors. This requires assessing expression levels from RNA obtained from the different tissues with microarrays, determining genes whose expression levels can be used as classifier variables, and then applying some rule to design the classifier from the sample microarray data. Design, performance evaluation, and application of classifiers must take into account randomness arising from both biological and experimental variability. To rapidly move from expression data to diagnostics that can be integrated into current pathology practice or to useful therapeutics, expression patterns must carry sufficient information to separate sample types.

Classification using a variety of methods has been used to exploit the class-separating power of expression data in cancer: leukemias [21], various cancers [22], small, round, blue-cell cancers [23], hereditary breast cancer [24], colon cancer [25], breast cancer [4], melanoma [26], and glioma [27].

Three critical statistical issues arise for expression-based classification [28, 29]. First, given a set of variables, how does one design a classifier from the sample data that provides good classification over the general population? Second, how does one estimate the error of a designed classifier when data is limited? Third, given a large set of potential variables, such as the large number of expression level determinations provided by microarrays, how does one select a set of variables as the input vector to the classifier? The problem of small-sample error estimation impacts variable selection in a devilish way. An error estimator may be unbiased but have a large variance, and therefore often be low. This can produce a large number of gene (variable) sets and classifiers with low error estimates. For a small sample, one can end up with thousands of gene sets for which the error estimate from the data at hand is zero. In the other direction, a small sample size enhances the possibility that a designed classifier will perform worse than the optimal classifier. Combined with a high error estimate, the result will be that many potentially good diagnostic gene sets will be pessimistically evaluated.

Not only is it important to base classifiers on small numbers of genes from a statistical perspective, but there are also compelling biological reasons for small classifier sets. As previously noted, correction of an aberrant function would be accomplished by the use of drugs. Sufficient information must be vested in gene sets small enough to serve as either convenient diagnostic panels or as candidates for the very expensive and time-consuming analysis required to determine

if they could serve as useful targets for therapy. Small gene sets are necessary to allow construction of a practical immunohistochemical diagnostic panel. In sum, it is important to develop classification algorithms specifically tailored for small samples [27].

While clustering algorithms do not produce the specificity and quantitative predictability of classification procedures, they can provide the means to group expression patterns that are coexpressed over a range of experiments in order to detect common regulatory motifs in an unsupervised manner. Moreover, by considering expression profiles over various tissue samples, clustering these samples based on the expression levels for each sample helps to develop techniques that offer the potential to discriminate pathologies and to recognize various forms of cancers or cell types. Clustering constitutes a supporting methodology for classification and prediction.

Many clustering approaches, such as K-means [30], self-organizing maps [31], hierarchical clustering [32], and others, have been applied to gene expression data analysis. One difficulty is that the selection of various algorithm parameters and other choices (e.g., type of linkage), initial conditions, and distance measures can all critically impact the results of clustering. Moreover, the number of clusters must often be chosen in advance. Therefore, comparison of results and analysis of the inference capability of clustering algorithms is important [33]. A good overview of clustering algorithms, as applied to gene expression data, including cluster validation, is available in [34].

4.2. Networks

A model of a genetic regulatory network is intended to capture the simultaneous dynamical behavior of all elements, such as transcript or protein levels, for which measurements exist. Needless to say, it is possible to devise theoretical models, for instance based on systems of differential equations, that are intended to represent as faithfully as possible the joint behavior of all of these constituent elements. The construction of the models, in this case, can be based on existing knowledge of protein-DNA and protein-protein interactions, degradation rates, and other kinetic parameters. Additionally, some measurements focusing on small-scale molecular interactions can be made, with the goal of refining the model. However, global inference of network structure and fine-scale relationships between all the players in a genetic regulatory network is still an unrealistic undertaking with existing genome-wide measurements produced by microarrays and other high-throughput technologies.

Thus, if we take the pragmatic viewpoint that models are intended to predict certain behavior, be it steady-state expression levels of certain groups of genes or simply the functional relationships between a group of genes, we must then develop them with the awareness of the types of data that are available. For example, it may not be prudent to attempt inferring dozens of continuous-valued rates of change and other parameters in differential equations from only a few discrete-time measurements taken from a population of cells that may not be synchronized with respect to their gene activities (e.g., cell cycle) and with a limited knowledge and understanding of the sources of variation due to the measurement technology and the underlying biology. What we should rather strive for is obtaining the simplest model that is capable of "explaining" the data at some chosen level of "coarseness" (Ockham's Razor). That is, we must strike the right balance between goodness-of-fit and model complexity.

Recently, a new class of models, called probabilistic Boolean networks (PBNs), has been proposed for modeling gene regulatory networks [16]. PBNs inherently capture the dynamics of gene regulation and activity, are probabilistic in nature, thus being able to absorb some of the uncertainty intrinsic to the data, are rule-based, and can be inferred from gene expression data sets in a straightforward manner. This class of models constitutes a probabilistic generalization of the well-known Boolean network model [35]. The PBN can be constructed so as to involve many simple but good predictors of gene activity. Just as importantly, it can include the situation where the structure of the model network changes in accord with the activity of latent variables outside the model, in effect, thereby resulting in a model composed of a family of constituent classical Boolean networks [17].

4.2.1. Prediction

The study of gene interaction and the concomitant behavioral changes due to signals external to the genome itself fits into the classical theories of nonlinear filtering, stochastic control, and nonlinear dynamical systems. Central to both analysis and design is prediction. With microarray technology, the gene expression measurements compose a random vector over time. They have a stochastic nature on account of both inherent biological variability and experimental noise. Genetic changes over time concern this random vector as a temporal process. Questions regarding the interrelation between genes at a given moment of time concern this vector at that moment. Comparison of two cell lines, say tumorigenic and nontumorigenic, involves two random processes and their cross probabilistic characteristics.

The genome is not a closed system. It is affected by intracellular activity, which in turn is affected by external factors. At a very general level, we might represent the situation by a pair of vectors, X denoting the gene expression time process and Z being a vector of variables external to the genome, either cellular or otherwise. In any practical situation, these will only include variables that are observable, measurable, and of interest. In a laboratory setting, Z might be composed of several components decided upon by the experimenter. Ultimately, our concern is with temporal transitions of X, affected by both the current states of X and Z. The most critical problem is the prediction of X at a future time from a current observation of X and knowledge of Z.

A predictor must be designed from data, which ipso facto means that it is an approximation of the predictor whose action one would actually like to model. The precision of the approximation depends on the design procedure and the sample size. Even for a relatively small number of predictor genes, good design can require a very large sample; however,

one typically has a small number of microarrays. There is also the computational problem inherent in the vast number of possible combinations of genes that can be involved in prediction. The problems of classifier design apply essentially unchanged when inferring predictors from sample data. To be effectively addressed, they need to be approached within the context of constraining biological knowledge, since prior knowledge significantly reduces the data requirement.

Even in the context of limited data, there are modest approaches that can be taken. One general statistical approach is to discover associations between the expression patterns of genes via the coefficient of determination [36, 37, 38]. This coefficient measures the degree to which the transcriptional levels of an observed gene set can be used to improve the prediction of the transcriptional state of a target gene relative to the best possible prediction in the absence of observations. The method allows incorporation of knowledge of other conditions relevant to the prediction, such as the application of particular stimuli or the presence of inactivating gene mutations, as predictive elements affecting the expression level of a given gene. Using the coefficient of determination, one can find sets of genes related multivariately to a given target gene. No causality is inferred. It may be that the target is controlled by a function of the predictive genes, or they predict well the behavior of the target because it is a switch for them. The relationship may involve intermediate genes in a complex pathway.

Another approach for finding groups of genes or factors that are likely to determine the activity of some target gene is the minimal description length (MDL) principle, which has been applied in the context of gene expression prediction [39]. This approach essentially seeks flexible classes of models with good predictive properties and considers the complexity of the models as a penalizing factor. With the fundamental goal being to improve the predictive accuracy or generalizability of the model [40], the MDL principle attempts to select the model that achieves the shortest code length describing both the data and the model. A related approach, called normalized maximum likelihood (NLM), has also been recently used for gene-expression-based prediction and classification [41].

4.2.2. Intervention

One reason for studying regulatory models is to develop intervention strategies to help guide the time evolution of the network towards more desirable states. Three distinct approaches to the intervention problem have been considered in the context of probabilistic Boolean networks by exploiting their Markovian nature. First, one can toggle the expression status of a particular gene from ON to OFF or vice versa to facilitate transition to some other desirable state or set of states. Specifically, by using the concept of the mean first passage time, it has been demonstrated how the particular gene, whose transcription status is to be momentarily altered to initiate the state transition, can be chosen to "minimize" in a probabilistic sense the time required to achieve the desired state transitions [42]. A second approach has aimed at changing the steady-state (long-run) behavior of the network by minimally altering its rule-based structure [43]. A third approach has focused on applying ideas from control theory to develop an intervention strategy, using dynamic programming, in the general context of Markovian genetic regulatory networks whose state transition probabilities depend on an external (control) variable [44].

5. CONCLUDING REMARKS

Computational genomics has been greatly influenced by data mining, partly due to the availability of large data sets and databases. Although data mining, as a discipline, is quite broad and lies at the intersection of statistics, machine learning, pattern recognition, and artificial intelligence, there are a number of challenging and important problems in computational genomics that can benefit from the application of engineering principles and methodologies, the latter being characterized by systems-level modeling and simulation.

Modern signal processing, though encompassing many of the same subject areas, has had a different history and background. As such, the applications around which the field has developed have been of a substantially different nature than those in data mining. While data mining problems are often centered around visualization and exploratory analysis of large high-dimensional data sets, finding patterns in data, and discovering good feature sets for classification, some common tasks in signal processing include removal of interference from signals, transforming signals into more suitable representations for various purposes, and analyzing and extracting some characteristics from signals.

Of importance in signal processing is the optimal design of operators under various criteria and constraints. That is, given a "true" signal and its noise-corrupted version, the goal is to find an optimal estimator, from some class of estimators (constraint), such that when it is applied to the noisy signal, some error (criterion) between its output and the true signal is minimized. Alternatively, if a representative signal is not available for training, armed with only the knowledge of the noise characteristics and a class of operators, the goal is to select an optimal estimator under a different criterion, such as minimizing the variance of the noise at its output.

Though these approaches have much in common with machine learning and statistical estimation theory, the nature of the constraints and criteria, and consequently the ensuing theory and algorithms, are guided by application-specific needs, such as detail and edge preservation, robustness to outliers, and other statistical and structural constraints. At the same time, much of the theory behind signal processing, in particular nonlinear digital filters, is tightly intertwined with dynamical systems theory, involving constructs such as finite and cellular automata.

It is clear that signal processing theory, tools, and methods can make a fundamental contribution to gene-expression-based classification and network modeling. Needless to say, traditional signal processing approaches, such as transform theory, can play an important role in other genomic applications, such as DNA or protein sequence analysis [45, 46, 47]. It is our belief that researchers with a background in

signal processing have the potential to make significant contributions and bring their unique perspectives to this exciting and important field.

REFERENCES

[1] G. Evan and T. Littlewood, "A matter of life and cell death," *Science*, vol. 281, no. 5381, pp. 1317–1322, 1998.

[2] J. L. DeRisi, L. Penland, P. O. Brown, et al., "Use of a cDNA microarray to analyse gene expression patterns in human cancer," *Nature Genetics*, vol. 14, no. 4, pp. 457–460, 1996.

[3] J. L. DeRisi, V. R. Iyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, no. 5338, pp. 680–686, 1997.

[4] C. M. Perou, T. Sorlie, M. B. Eisen, et al., "Molecular portraits of human breast tumours," *Nature*, vol. 406, no. 6797, pp. 747–752, 2000.

[5] L. Wodicka, H. Dong, M. Mittmann, M. H. Ho, and D. J. Lockhart, "Genome-wide expression monitoring in *Saccharomyces cerevisiae*," *Nature Biotechnology*, vol. 15, no. 12, pp. 1359–1367, 1997.

[6] H. H. McAdams and L. Shapiro, "Circuit simulation of genetic networks," *Science*, vol. 269, no. 5224, pp. 650–656, 1995.

[7] C.-H. Yuh, H. Bolouri, and E. H. Davidson, "Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene," *Science*, vol. 279, no. 5358, pp. 1896–1902, 1998.

[8] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[9] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young, "Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks," in *Proc. 6th Pacific Symposium on Biocomputing*, pp. 422–433, Mauna Lani, Hawaii, USA, January 2001.

[10] E. J. Moler, D. C. Radisky, and I. S. Mian, "Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in *S. cerevisiae*," *Physiological Genomics*, vol. 4, no. 2, pp. 127–135, 2000.

[11] K. Murphy and S. Mian, "Modelling gene expression data using dynamic Bayesian networks," Tech. Rep., Computer Science Division, University of California, Berkeley, Calif, USA, 1999.

[12] M. Wahde and J. A. Hertz, "Coarse-grained reverse engineering of genetic regulatory networks," *Biosystems*, vol. 55, pp. 129–136, 2000.

[13] D. C. Weaver, C. T. Workman, and G. D. Stormo, "Modeling regulatory networks with weight matrices," in *Proc. Pacific Symposium on Biocomputing*, vol. 4, pp. 112–123, Mauna Lani, Hawaii, USA, January 1999.

[14] T. Mestl, E. Plahte, and S. W. Omholt, "A mathematical framework for describing and analysing gene regulatory networks," *Journal of Theoretical Biology*, vol. 176, no. 2, pp. 291–300, 1995.

[15] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[16] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[17] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[18] A. Arkin, J. Ross, and H. H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells," *Genetics*, vol. 149, no. 4, pp. 1633–1648, 1998.

[19] R. Somogyi and L. D. Greller, "The dynamics of molecular networks: applications to therapeutic discovery," *Drug Discovery Today*, vol. 6, no. 24, pp. 1267–1277, 2001.

[20] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, no. 4, pp. 364–374, 1997.

[21] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[22] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 559–583, 2000.

[23] J. Khan, J. S. Wei, M. Ringner, et al., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, no. 6, pp. 673–679, 2001.

[24] I. Hedenfalk, D. Duggan, Y. Chen, et al., "Gene-expression profiles in hereditary breast cancer," *New England Journal of Medicine*, vol. 344, no. 8, pp. 539–548, 2001.

[25] U. Alon, N. Barkai, D. A. Notterman, et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

[26] M. Bittner, P. Meltzer, J. Khan, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.

[27] S. Kim, E. R. Dougherty, I. Shmulevich, et al., "Identification of combination gene sets for glioma classification," *Molecular Cancer Therapeutics*, vol. 1, no. 13, pp. 1229–1236, 2002.

[28] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York, NY, USA, 1996.

[29] E. R. Dougherty, "Small sample issues for microarray-based classification," *Comparative and Functional Genomics*, vol. 2, no. 1, pp. 28–34, 2001.

[30] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nature Genetics*, vol. 22, no. 3, pp. 281–285, 1999.

[31] P. Tamayo, D. Slonim, J. Mesirov, et al., "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 6, pp. 2907–2912, 1999.

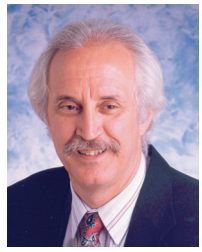
[32] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 25, pp. 14863–14868, 1998.

[33] E. R. Dougherty, J. Barrera, M. Brun, et al., "Inference from clustering: application to gene-expression time series," *J. Comput. Biol.*, vol. 9, no. 1, pp. 105–126, 2002.

[34] Y. Moreau, F. de Smet, G. Thijs, K. Marchal, and B. de Moor, "Functional bioinformatics of microarray data: from expression to regulation," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1722–1743, 2002.

[35] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York, NY, USA, 1993.

[36] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.
[37] S. Kim, E. R. Dougherty, M. L. Bittner, et al., "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *Biomedical Optics*, vol. 5, no. 4, pp. 411–424, 2000.
[38] S. Kim, E. R. Dougherty, Y. Chen, et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.
[39] I. Tabus and J. Astola, "On the use of MDL principle in gene expression prediction," *EURASIP Journal on Applied Signal Processing*, vol. 2001, no. 4, pp. 297–303, 2001.
[40] I. Shmulevich, "Model selection in genomics," *EHP Toxicogenomics*, vol. 111, no. 6, pp. A328–A329, 2003.
[41] I. Tabus, J. Rissanen, and J. Astola, "Normalized maximum likelihood models for Boolean regression with application to prediction and classification in genomics," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 2002.
[42] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene Perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.
[43] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," *Journal of Biological Systems*, vol. 10, no. 4, pp. 431–445, 2002.
[44] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning Journal*, vol. 52, no. 1-2, pp. 169–191, 2003.
[45] D. Anastassiou, "Frequency-domain analysis of biomolecular sequences," *Bioinformatics*, vol. 16, no. 12, pp. 1073–1081, 2000.
[46] P. D. Cristea, "Large scale features in DNA genomic signals," *Signal Processing*, vol. 83, no. 4, pp. 871–888, 2003.
[47] K. M. Bloch and G. R. Arce, "Analyzing protein sequences using signal analysis techniques," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds., pp. 113–124, Kluwer Academic Publishers, Boston, Mass, USA, 2002.



Edward R. Dougherty is a Professor in the Department of Electrical Engineering at Texas A&M University in College Station. He holds an M.S. degree in computer science from Stevens Institute of Technology in 1986 and a Ph.D. degree in mathematics from Rutgers University in 1974. He is the author of eleven books and the editor of other four books. He has published more than one hundred journal papers, is an SPIE Fellow, and has served as an Editor of the Journal of Electronic Imaging for six years. He is currently Chair of the SIAM Activity Group on Imaging Science. Prof. Dougherty has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His current research focuses on genomic signal processing, with the central goal being to model genomic regulatory mechanisms. He is Head of the Genomic Signal Processing Laboratory at Texas A&M University.

Ilya Shmulevich received his Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, Ind, USA, in 1997. From 1997 to 1998, he was a Postdoctoral Researcher at the Nijmegen Institute for Cognition and Information at the University of Nijmegen and National Research Institute for Mathematics and Computer Science at the University of Amsterdam in the Netherlands, where he studied computational models of music perception and recognition. From 1998 to 2000, he worked as a Senior Researcher at Tampere International Center for Signal Processing in the Signal Processing Laboratory at Tampere University of Technology, Tampere, Finland. Presently, he is an Assistant Professor at Cancer Genomics Laboratory at The University of Texas MD Anderson Cancer Center in Houston, Tex. He is an Associate Editor of Environmental Health Perspectives: Toxicogenomics. His research interests include computational genomics, nonlinear signal and image processing, computational learning theory, and music recognition and perception.

Michael L. Bittner was initially trained as a biochemical geneticist, studying phage replication and bacterial transposition with a variety of biochemical and bacterial genetic methods at Princeton University, where he received his Ph.D. degree from Washington University School of Medicine, and the Population and Molecular Genetics Department of the University of Georgia, where he carried out his postdoctoral researches. Since that time, his efforts was concentrated on the practical application of knowledge about the control systems operating in prokaryotes and eukaryotes. At Monsanto Corporation in St. Louis, Dr. Bittner was involved in developing technology for the biologic production of peptides and proteins useful in human medicine and agriculture. At Amoco Corporation in Downers Grove, Illinois, he played a central role in developing methods for producing, in yeast, small molecule precursors of vitamins of human and veterinary pharmacologic interest. He collaborated in the development of cytogenetic molecular diagnostics based on in-situ hybridization that produced a series of technologies leading to the founding of Vysis Corporation, also in Downers Grove. His recent efforts in the National Institutes of Health and the Translational Genomics Research Institute focus on developing ways of making accurate measures of the transcriptional status of cells and analytic tools that allow inferences to be drawn from these measures that provide insight into the cellular processes operating in healthy and diseased cells.