Applications of Signal Processing to Genetics

Matthew J. Berryman, *Student Member, IEEE,* Douglas A. Gray, *Member, IEEE,* Christopher Wilkinson, Andrew Allison, *Member, IEEE,* and Derek Abbott, *Senior Member, IEEE*

***Abstract*— This paper reviews applications of signal processing techniques to a number of areas in the field of genetics. We focus on techniques for analyzing DNA sequences, and briefly discuss applications of signal processing to DNA sequencing that determines the sequences, and other related areas in genetics that can provide biologically significant information to assist with sequence analysis.**

I. Introduction

Genetics is concerned with the physical characteristics of organisms that are passed on from one organism to another through the use of deoxyribonucleic acid (DNA). DNA consists of a sequence of nucleotides. The nucleotides are the chemical bases adenosine, thymine, cytosine and guanine that are denoted using the alphabet $\{\mathrm{A}, \mathrm{T}, \mathrm{C}, \mathrm{G}\}$. Those on one strand are paired in a complementary fashion with those on the other strand, where adenosine matches with thymine, and guanine with cytosine. Groups of three bases are called codons, and these encode the twenty amino acids which combine to form proteins, the building blocks of life. Not all regions of DNA code for proteins though, some of these non-protein-coding regions have known functions, such as the *Xist* gene [1], which codes for an ribonucleic acid (or RNA) molecule that deactivates one of the two X chromosomes in female mammals. These RNAs may play an important role in the complexity of organisms such as humans [2]. There are also promoter regions around genes which act as targets for gene activation or deactivation [3]. Other non-coding regions appear to only be “junk” DNA left over from the biological past, with little or no use – or perhaps have a yet undiscovered function. Biologists have suggested that “junk” regions may act as a form of isolation between coding regions and may also act as error-robust locations for sexual recombination – this is described further in Harmer *et al.* [4], where it is conjectured that these effects could be modeled in game-theoretic terms.

Signal processing is the use of mathematical techniques to analyze any data signal. This data could be an image, a sound, or any other sequence of data like in a gene. Signal processing techniques are used in the field of genetics research to detect, estimate, and classify features of interest in the data, and are also used in DNA sequencing [5], [6], which produces the many DNA sequences of genomes available on the Internet. This data, and data generated by other techniques like the DNA fingerprinting used by crime authorities, is also amenable to analysis by signal processing techniques.

Another area where signal processing techniques have enjoyed wide usage is in microarray processing [7]. In microarray analysis, effects on gene expression can be tested, for example the effect of a drug. Microarrays are a colored grid of spots (typically one color for the control, the other for the cells under test) with spot intensity and color showing the expression levels for the gene associated with that spot.

The analysis of the sequences produced has come under intense focus as an area where signal processing techniques could be used to solve a number of important problems such as the nature of non-coding DNA and distinguishing coding DNA from non-coding DNA. Methods such as the discrete Fourier transform [8], [9] and multifractal techniques [10] have been applied to the problem, complementing more traditional techniques that often use hidden Markov models [11], [12]; these are detailed later. A good overview of Fourier transform methods and wavelet transforms not discussed in this paper and a more in depth discussion of cellular neural networks can be found in Zhang *et al.* [13]. Here we focus on other applications of Fourier methods, and also explore the use of hidden Markov models and other mathematical techniques to general problems in genetics.

II. DNA sequencing

A. Introduction to DNA sequencing

For a section of DNA to be sequenced, copies of that DNA are broken up into fragments of varying length, all with the same starting point but with a random end point. The end points are then labelled with a fluorescent dye, and the fragments are then run along a capillary tube, the velocity of the sequences varying proportionally to their length. A laser or other scanning device then reads the set of four color intensities, one for each base. More details of this technique can be found in Mastrangelo *et al.* [14]. A peak in one of the color signals at a particular position indicates the respective base that occurs at that postion. The task is not as easy as it might first appear, due to inherent problems in the technique, some of which can often be resolved using signal processing techniques. An example of a sequence of peaks, with the given sequence determined using signal processing techniques is shown in Figure 1.

B. Current signal processing techniques for sequencing

A widely used piece of software used in sequencing is *Phred* [5], [6], which uses Fourier techniques (see Appendix A) in producing a sequence of bases from a set of varying color intensities. Techniques used in other sequencing software

Department of Electrical and Electronic Engineering, and Centre for Biomedical Engineering, The University of Adelaide, SA 5005, Australia (email: {mattjb,dagray,aallison,dabbott}@eleceng.adelaide.edu.au)

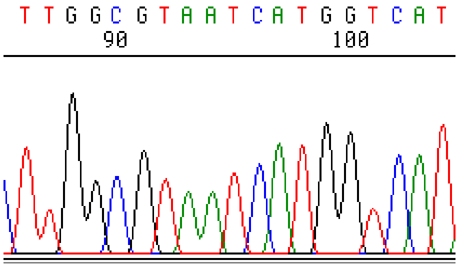


Fig. 1. This figure shows four color signals representing the intensity of the colors at different positions. The signals come from scanning a set of phosphor tagged fragments of DNA of various lengths being run down a gel. The determined sequence of bases, from is shown on top, along with the position in the sequence. The presence of a red signal at a certain position indicates a T, black for G, blue for C, and green for A, these are not necessarily the colors used in the actual tags.

include maximum likelihood [15] and Bayesian analysis [16]. *Phred* has four main phases of operation:

1) Locating predicted peaks.
2) Locating observed peaks.
3) Matching observed and predicted peaks.
4) Finding peaks missed by step three.

Note that the phases operate on each of the four traces separately. A signal flow diagram showing the flow of data through each of the phases is given in Figure 2.

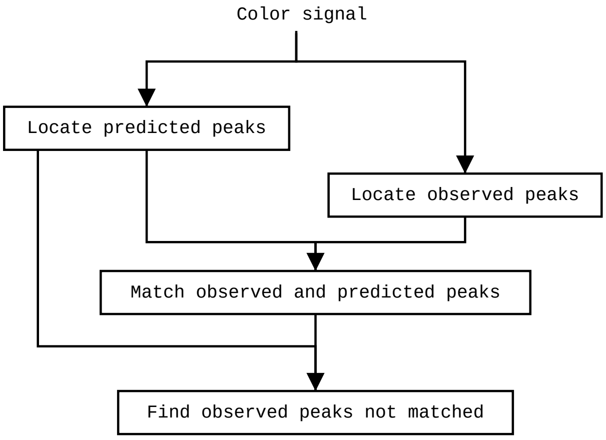

Color signal
Locate predicted peaks
Locate observed peaks
Match observed and predicted peaks
Find observed peaks not matched

Fig. 2. This shows the order of the four phases used in *Phred* to process the DNA signal trace data, and the flow of data between the four phases.

In phase one, a frequency modulated square wave is overlayed over easily detectable peaks. The square wave is then used to find areas where the variation in inter-peak spacing is low. Starting with the region (a window of 200 points) with the lowest variation in inter-peak spaces, and moving outwards to both ends of the sequence, the software then overlays a set of sine curves over the trace. It finds the best ones using Fourier techniques that match sine waves to the peaks. This phase then gives a set of predicted locations.

In phase two, a set of observed peaks are found by scanning the traces for concave regions, that is, for a point i in the trace where $2v(i) \geq v(i+1) + v(i-1)$, for $v(k)$ the value of the trace at point k. The observed peak position is then given by the median of the area under the peak, provided the total area of the peak meets certain specifications when compared to the surrounding peaks. Because, in phase three, the peak may need to be split into virtual peaks of which one will match a predicted peak, the locations of points which split the area into two, three, or four equal area regions are also determined.

In phase three, three stages are used to find easy matches, align observed and predicted peaks that are not easy matches, and to find more difficult matches. All three algorithms work with relative areas of peaks and the shifts required to match the observed peaks to the predicted peaks. Those not successfully matched at each stage are passed on to the further stages which take longer to run, thus the software uses the fastest algorithm for each peak as appropriate.

The letter N is used in the sequence string (which would ideally consist of the letters A,T,C, and G) to denote a base where an observed peak cannot be associated with a predicted peak.

Occasionally, due to problems in the physical processes that generate the traces, there are some well-resolved observed peaks that remain unmatched due to an underestimation of predicted peaks in a region. These peaks are recovered if they meet all of the following criteria:

(i) the observed peak has the largest of the four signals at its time point,
(ii) meets the minimum size criterion used in phase two,
(iii) is unsplit,
(iv) is flanked by resolved peaks, and
(v) is such that adding it results in improved peak spacing.

Each base in the finished sequence is then given an error score, which denotes the probability of error: a score of x denotes a probability of error of $10^{-x/10}$, so a score of 50 for a particular base indicates the probability that it has been called in error is 1 in $100,000$.

As the sequencing method described above only works with relatively short sequences of DNA (around 500 bases long) from a much larger sequence (the human genome is around three billion bases long), a program to reassemble the large sequence from the smaller sequences produced by *Phred* is needed. The *Phrap* software package is used in conjunction with the *Consed* software to reassemble the sequence automatically (the role of *Phrap*) and manually given quality scores (*Consed*) [6].

Further discussion of the algorithms and issues of sequence assembly can be found in Pop *et al.* [17].

C. Future directions

With improvements in robotics, computers, and computer algorithms, the ability to sequence large amounts of DNA will improve dramatically. The existing techniques used above will still be useful, however other algorithms developed may prove faster and more useful.

III. Sequence Analysis

Once a sequence has been obtained, one can then use it to answer questions about DNA and carry out biological analysis *in silico* [1]. Some of the many things that can be determined about the sequence are:

[1] *in silico* refers to a biological "experiment" done in software.

1) where the genes are located
2) how the proteins encoded by the genes “fold” into three dimensional structures
3) the relationships between genes in different organisms
4) searching sequences for genes related to known ones
5) examining lateral gene transfer (where genes are transfered between existing species)
6) correlations between regions of DNA.

Current techniques mainly use statistical and probabalistic techniques, especially hidden Markov models [11]. Recently, others have considered applying signal processing techniques [8] and fractal techniques [10] to these problems.

A. Current techniques

Many of the existing techniques for solving problems like finding the position of genes and determing protein folding are based around hidden Markov models. Hidden Markov models are statistical models for describing events in a given state-space, and act as a mathematical profile of the sequence, capturing important details. Hidden Markov models are trained on a set of data, with some assumptions about the data built in to the algorithm. Once trained, the model can then take a new sequence and find genes in it, or determine the way the encoded protein folds, or look for similar sequences in a larger new sequence in a computationally efficient way. Details of the training of hidden Markov models are given in Appendix B.

Essentially Markov models use a state-based approach to examine sequences, with a set of probabilities giving the probability of the system changing from one state to another. For example, a simple Markov model might treat a base as a state, and determine the probability that a T occurs after a G in the sequence. A *hidden* Markov model considers sets of states which are not directly observable in a sequence, for example the *GeneMark.hmm* software [12] has separate sets of states for coding and non-coding regions of DNA, so an A in a coding region is a different state to an A in a non-coding region. Note that the coding and non-coding regions are not observable in the sequence by itself, so a hidden Markov model is required in this case.

One application of hidden Markov models is in gene finding [12]. Here one takes a DNA sequence, just a long string letters from $\{\mathrm{A},\mathrm{T},\mathrm{C},\mathrm{G}\}$, and with no other information other than knowledge of the start and stop codons one can predict genes with a missed gene rate (when the predicted genes are compared to known genes) of around 5%. Hidden Markov models have also been used to predict protein folding for the proteins encoded by known genes [18], with prediction of various structures within proteins having an accuracy around $55-70\%$ after being trained on known protein structures. Other related statistical modelling techniques can give accuracy rates up to 77% [19].

It is often of interest to build up profiles of biological sequences (both sequences of bases and sequences of amino acids), to enable comparisons of sequences between species, within species and comparisons between related sequences within an individual genome. Software is available that lets the user build a database of profiles, which can then be used for the above mentioned purposes [20]. Using this software, for example, one can build a hidden Markov model profile of the *thrA* gene in the *Escherichia coli* strains *E. coli* K12 and *E. coli* O157:H7 EDL933 [21], and then use this to find and align the same gene in the CFT073 strain [22]. Searching and aligning can be done with other algorithms, such as the Smith-Waterman algorithm [11], [23]. Here we show the match of part of the gene sequence found in CFT073. The first line gives a part of the sequence in the trained hidden Markov model. The third line, in upper case, gives part of the query sequence which matches the model; the matches of individual bases are shown in the second line along with the differences as indicated by gaps. The query sequence is usuall shown in uppercase to distinguish it from the model and match sequences.

```
model     ccacctggtggcg
matches   cca ctggt gcg
query     CCATCTGGTAGCG
```

The match was found by using a hidden Markov model, which finds matched states that clearly are not directly observable in the sequence itself, being merely just a string of bases.

B. Spectra and correlations

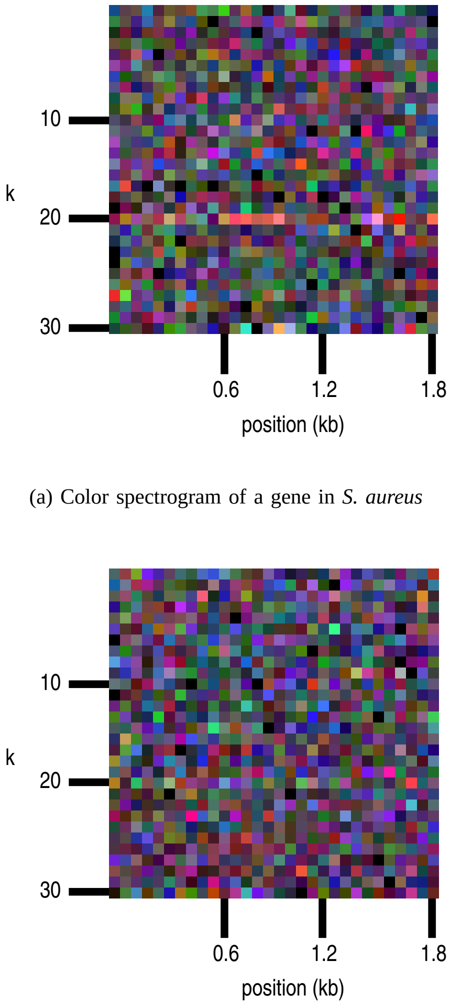
An application of the discrete Fourier transform (see Eq. 9), as given in Anastassiou [8] is to generate color spectrograms, which enable the visual identification of regions in DNA where sequences are repeated, and what the repeat length is. Fourier transforms can also be used to find genes [24], [25]. To illustrate the usefulness of the color spectrogram technique in identifying regions of DNA, Figure 3 shows the color spectrograms of DNA sequences in *Staphylococcus aureus Mu50* [26] and *Homo sapiens* [27]. To obtain the color spectrograms the DNA sequence is converted to sequences of numbers and as described further in Anastassiou [8] and the methods described in Appendix A are applied to this sequence. The presence of codons (length $T=3$) shows up as a bright band at discrete frequency $k=N/T$ where N is the sequence length. Since we use subsequences of length 60 this band appears at discrete frequency $k=60/3=20$.

C. Generalized correlation detection

The power spectrum is related to the autocorrelation of a sequence by the Weiner-Khintchine relationship

$$P(f)=\frac{1}{N}\sum_{i=1}^{N}R_{ss}(i)e^{-j2\pi if}, \qquad (1)$$

where N is the data length, $j=\sqrt{-1}$, and $R_{ss}(i)$ is the autocorrelation of the sequence for sequence distance i. This is only exactly true if the DNA sequence is a stationary process. Drifts in GC content throughout a genome mean this is not true in general across a genome [27]. There are a variety of other methods for analyzing correlations in DNA, from mutual information [28] to correlation functions [29] and fractal techniques such as the Higuchi method [30] and those discussed below. The mutual information measure, as given in



(a) Color spectrogram of a gene in *S. aureus*

(b) Color spectrogram of the *BRCA2* gene in *H. sapiens*

Fig. 3. Figure 3(a) shows the color spectrogram for a 1.8 kilobase (kb) region in the *gyrA* gene of *S. aureus Mu50*. The 1.8 kb sequence is broken down into sequences of length $N = 60$, each sequence is transformed into three different sequences according to a transformation given in Anastassiou [8]. The total power in the three sequences then gives a set of RGB color intensities, and the corresponding color plotted for discrete frequencies k running down each column of pixels, each column corresponding to a sequence of length 60. A bright band is visible in Figure 3(a), but not in Figure 3(b), indicating a lack of codons in the human DNA due to introns.

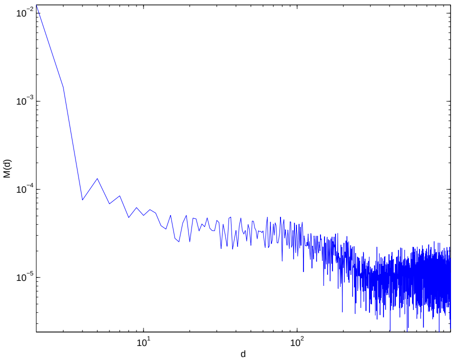


Fig. 4. This figure shows the mutual information plot of *E. coli* K12. The values of mutual information over the sets of bases separated by distance d were computed using Eq. 20, for d up to 1000. Note that significant correlations exist only up to a few hundred bases.

Appendix C, can be used to measure the mutual information (and hence correlations) across an entire genome [31], [32], an example of the use of this on the *Escherichia coli* K12 genome [21] is shown in Figure 4. An excellent overview of these and other techniques for studying correlations in DNA, and the implications of the results obtained using these methods can be found in a paper by Li [33].

Correlations can arise as a result of genetic processes such as gene duplication and insertions [34], and these techniques can provide indication of such events, as well as other structures present in DNA sequences. Another technique used as part of correlation and structure detection is the DNA walk technique [35], [36]. Figure 5(a) shows the result of doing a “walk”, where a step downwards is taken if a G or C is encountered in the sequence. A related technique is to map the bases onto complex numbers, and plot the cumulative phase [37]. We use the mapping for sequence element $s(i)$ given in Eq. 2,

$$\phi(i) = \begin{cases} \pi/4, & s(i) = A, \\ 3\pi/4, & s(i) = T, \\ -\pi/4, & s(i) = C, \\ -3\pi/4, & s(i) = G. \end{cases} \tag{2}$$

If the bases are evenly distributed, this gives an average phase of 0, and the mapping is designed to highlight the GC content similar to the DNA walk. A phase plot is shown Figure 5(b).

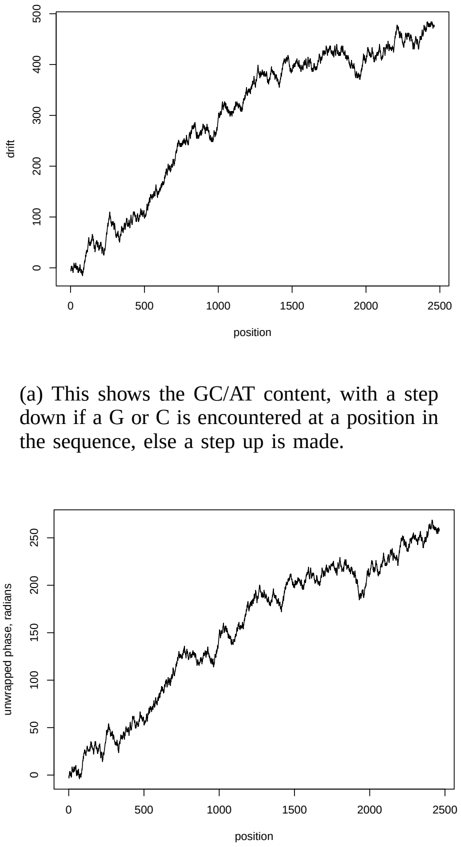
D. Linguistics

Since DNA and amino acid sequences can be thought of as a type of language, it makes sense to try and use techniques from computational linguistics to analyze genetic sequences. This theory of grammar in a computational sense was first developed by Chomsky [38], [39]. It has been applied to a wide range of applications in sequence analysis from determining gene structures [40] to RNA (ribonucleic acid) secondary structure [41]. Mantegna *et al.* have taken methods from statistical linguistics, along with information theory approaches, to consider differences between non-coding and coding DNA [42]–[44]. This reveals the presence of hidden information and extra redundancy in non-coding regions, perhaps due to lengthy promoter regions [45], or due to information left from former coding regions. A good overview of linguistic techniques used can be found in Durbin *et al.* [11].

The PROSITE database contains a large number of protein families (related sequences), and their patterns, or “motifs” [46], [47]. This database can be searched using PROSITE patterns, an example of a pattern is

```
[ACFI]-[QC]-G-[AF]
```

where the capital letters denote amino acids. Square brackets denote that any one of the enclosed amino acids can occur in that position in the matched sequence, curly brackets (not used here) denote that none of the enclosed amino acids can occur



(a) This shows the GC/AT content, with a step down if a G or C is encountered at a position in the sequence, else a step up is made.

(b) This shows the cumulative phase, with the phase added at each position determined by Eq. 2. Note the similarity to Figure 5(a) due to the mapping used, but note that extra information is evident in the region from position 800 to position 1000.

Fig. 5. Two techniques have been used to show the structure in the DNA sequence of the *thrA* gene in *E. coli*. This sequence is clearly AT-rich, indicated by the upward trend of both graphs

in that position in the matched sequence. This can be written as a set of regular grammar rules, starting with position S and with W_i the positions of the sequence,

$$\begin{array}{rcl} S & \rightarrow & AW_1|CW_1|FW_1|IW_1 \\ W_1 & \rightarrow & QW_2|CW_2 \\ W_2 & \rightarrow & GW_3 \\ W_3 & \rightarrow & A|F, \end{array} \tag{3}$$

where $\rightarrow$ means “rewrite as” and $|$ means “or”. The fact that PROSITE patterns can be written as regular grammars means the searching for sequences which contain the motif is highly efficient [11].

A new approach to feature detection in language is based upon inter-word spacing [48], or better referred to in a language context as word recurrence interval (WRI) [49], which is the number of words between each occurrence of a particular word. In DNA one would consider inter-oligomer spacing. This has potential applications to classifying organisms [49], however to use this method on DNA and protein sequences one would have to define what a “word” is – for example, is it a gene or an exon? A method based on large scale structures like WRI, called gene order conservation, has shown some promise [50].

E. *Information theory and fractals*

Other techniques with possible applications in the area of sequence analysis include information theory approaches [51], and related fractal approaches [52]. Multifractal approaches can be used to classify bacteria by a just a few numbers derived from the whole genome DNA sequence. Obviously this has limited use because of the large number of places in the genome sequence that even closely related species differ by, however the multifractal technique has shown some promise in general categorization of bacteria [52]. Other information theory approaches have been used in areas such as binding site recognition [51]. Phylogenetic trees are constructed from genetic data, and show the relationship between organisms based on their genetic data. Figure 6 shows two phylogenetic trees, one a known tree, and the other a tree constructed using a multifractal distance measure.

IV. IMAGE PROCESSING

A. *Types of images*

There are a number of different types of images that are analyzed by biologists other than the sequencing and microarray images as discussed elsewhere in this paper. One common type of image that is generated is where cells are tagged with a flourescent dye according to the protein expression level of a particular gene. Different color dyes are used to indicate the expression levels of different genes. Figure 7 shows the expression levels of the *MTA1* and *ER*-α genes in cells from *Mus musculus* [53].

Another class of images are the electrophoresis images. These are similar to the sequencing images in Section II, except instead of scanning the DNA fragments from one organism moving past a scanner, there are DNA fragments from several organisms (or parts of organisms) moving along different tracks, and an X-ray image is taken of all the tracks at a point in time. This generates a grayscale image like that shown in Figure 8, which shows the expression of the *ADAM33* gene involved with asthma in types of human cells [54].

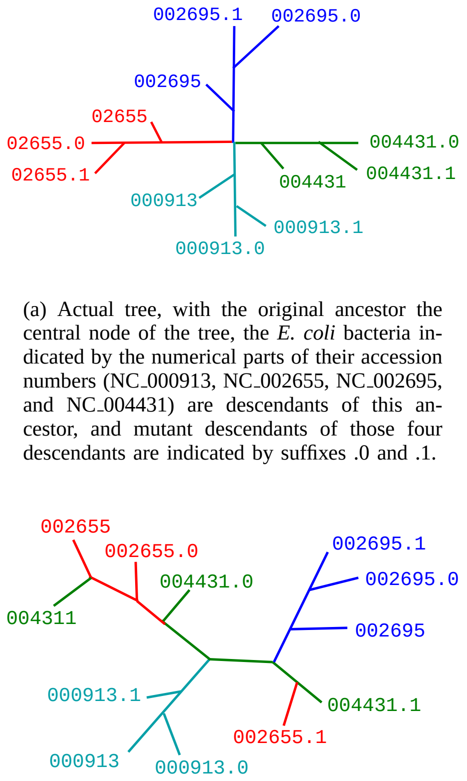
With these types of images one can use image processing techniques to:

- Identify regions of interest, using edge detection algorithms.
- Count these regions of interest.
- Determine the location of these regions.

B. *Image processing techniques and applications*

Here are the details of several of the edge detection algorithms used to find the regions, and a discussion of other processing techniques, which can be used in conjunction with these algorithms to provide useful results to geneticists.

Color filtering is a technique that can be used to extract only the colors of interest from images. Figure 9 shows how color filtering can highlight the cells tagged with red in Figure 7.



(a) Actual tree, with the original ancestor the central node of the tree, the *E. coli* bacteria indicated by the numerical parts of their accession numbers (NC_000913, NC_002655, NC_002695, and NC_004431) are descendants of this ancestor, and mutant descendants of those four descendants are indicated by suffixes .0 and .1.

(b) This is the tree of the same bacteria, generated using a multifractal measure of distance [52], and the neighbor joining algorithm [11], where a minimum distance means the two species are neighbors on the tree. The multifractal clearly has trouble distinguishing between such closely related species, but has some potential, and could be combined with other phylogenetic measures.

Fig. 6. These phylogenetic trees show the relationship between different organisms. Descendants of one organism are shown as branches from that organism, thus a family of related organisms with a common ancestor will all be on subtrees branching from the point at which that organism is shown.

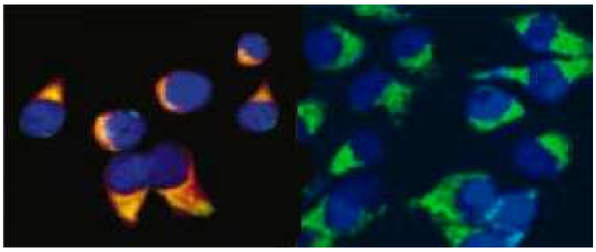


Fig. 7. This figure (from [53]) shows the expression of the *MTA1* (red) and *ER-α* (green) genes in mouse cells under different conditions. DNA is stained in blue, this helps to show the general outlines of the cells

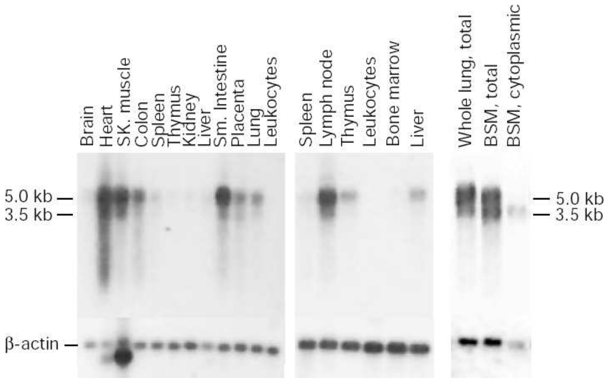


Fig. 8. This is an electrophoresis image (from [54]) showing the expression of the *ADAM33* gene in various types of human cells. The expression of *β-actin* is also shown, since all cells produce this in relatively similar amounts and so it acts as a control marker on the image. The 5.0 kb (kilobase) and 3.0 kb fragments are visible also, with shorter fragments traveling further down the gel.

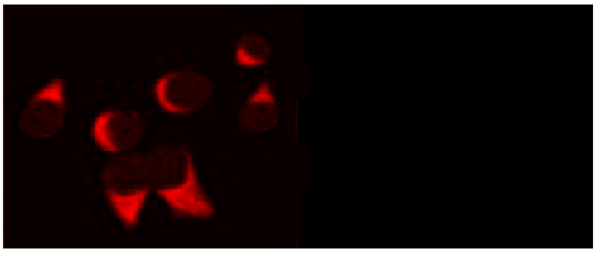


Fig. 9. This is the result of applying a color filter to the image in Figure 7 (from [53]). Note that there is still some background red color, but the red regions in the original image are much easier to distinguish.

Edge detection is the detection of edges of various objects within an image against some background. Methods for detecting edges range from simple thresholding to more complicated cross and Sobel algorithms [55].

Roberts' cross method uses the masks shown in Figure 5 [56]. An extension of Roberts' cross method is the Sobel operator [55]. This consists of the masks given in Equation 6. The masks are applied independently, using convolution, which is defined as

$$O(i,j) = \sum_{k=1}^{m} \sum_{l=1}^{n} I(i+k-1, j+l-1) K(k,l), \qquad (4)$$

for $I(i+k-1, j+l-1)$ the set of brightness values for the image pixels in an image I with M rows and N columns, $K(k,l)$ the values in the mask (also called a convolutional kernel) with m rows and n columns, and $O(i,j)$ the output, $1 < i < M - m + 1$ and $1 < j < N - n + 1$. Equation 5 shows the masks for Roberts' cross method, and Equation 6 shows the masks for the Sobel method,

$$K_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad K_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \qquad (5)$$

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \qquad (6)$$

The result of applying the Sobel operators gives a set of magnitudes

$$\sqrt{G_x^2 + G_y^2}, \tag{7}$$

and directions

$$\arctan\left(\frac{G_y}{G_x}\right), \tag{8}$$

where G_x and G_y are obtained from the values $O(i,j)$ with the kernel set first to the x-mask and then the Chy-mask respectively. The magnitudes and directions are then combined to give a set of direction vectors at each point in the output, some thresholding is then applied and the resultant values give an image that ideally shows only the edges. An example of applying the Sobel method to an image can be seen in Figure 10.

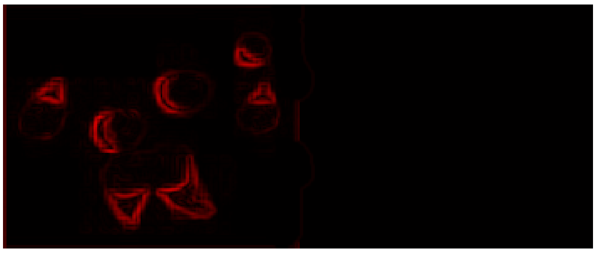


Fig. 10. This image is the result of applying the Sobel method to the filtered image shown in Figure 9 (from [53]). The edges of the bright red regions in Figure 9 can clearly be seen.

V. Microarray processing

A. Introduction to microarray technology

Microarrays, also known as gene or DNA chips, provide a relatively rapid way of analyzing gene expression patterns in an organism. Genes are expressed in different levels according to cell function, which may be altered in response to changes in its environment or may simply vary with time. The uses microarray technology are numerous, and include identification of complex genetic diseases, drug discovery, pathogen detection and analysis, and detecting different expression of genes over time. Further details on microarray technology and potential uses may be found in The Chipping Forecast (I and II) published free online by Nature Genetics [57], [58].

A microarray is an array of probes for detecting the expression levels of tens of thousands of genes simultaneously. In a typical two-color microarray experiment the relative expression levels between two target samples (cells) is measured for each probe. For each target, the messenger RNA (mRNA, the transcribed genes from the DNA) is used to form complementary DNA (cDNA), labeled with a particular color dye. Typically green and red dyes are used. The two target cDNA samples are then passed over the probes, and the target cDNA fragments bind (hybridize) to probes according to matching probe sequences. The microarray is then imaged using a laser scanner which measure the fluorescence intensities of each dye. The ratio of intensities for each probe is a measure of the relative abundance, and hence gene expression level, of the corresponding DNA sequence in the two samples. So if a colored spot is bright yellow (bright green plus bright red) it indicated both target samples have the gene corresponding to that spot highly expressed in equal amounts. See Figure 11 for an example of a microarray image from a two colour system [59], and Yang *et al.* for a more detailed description on the hybridisation and scanning procedure [60].

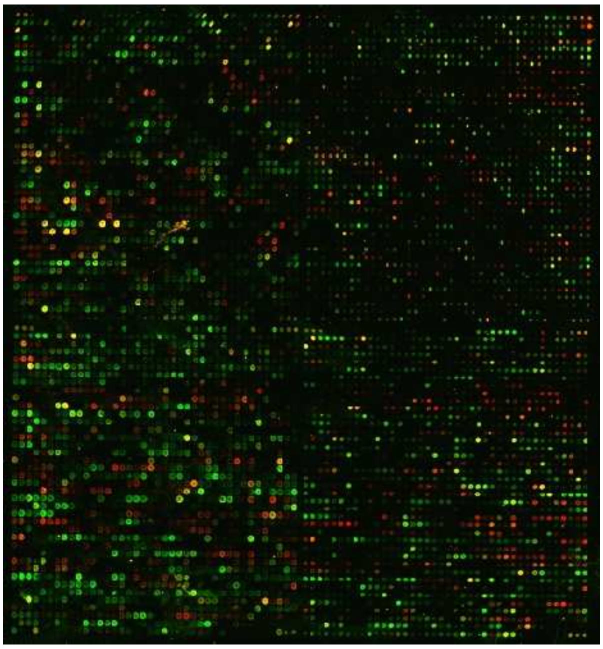


Fig. 11. A microarray showing gene expression for all 6000 yeast genes (from [59]), with the two sets of data coming from various points in the "diauxic shift", where the yeast go from fermentation (anaerobic respiration) to respiration (aerobic respiration).

Image analysis techniques are needed to analyze the data, both in aiding trained users in determining the results, and providing results to untrained users. Several different technologies exist for the printing of microarray slides. Two color microarrays are printed using robotic arrayers which deposit probe material from cDNA or oligonucleotides libraries onto the microarray slide, or a modified inkjet printer which builds oligonucleotide probes up base-pair by base-pair. Single colour microarrays are produced using propriety technologies based on photolithography or the digital light processor and feature significantly higher spot density than their two-colour relatives. The two different classes of microarray platform require different image analysis approaches. With a trend in minaturising the gene chips [14] combined with rapid image analysis of the results, it becomes possible to perform field tests for pathogens.

B. Current applications of signal processing to microarrays

The goal of processing microarrays is to turn the image of the array into a set of values giving the level of expression for each gene under analysis. The main tasks in processing this data are:

1) Clear the image from the background: the areas where spots do not occur will contain background noise color, and this needs to be cleaned up.

2) Spot detection: a grid is overlayed over the array of spots, and those spots not occuring clearly within the grid cells are deleted (as sometimes a blotch will occur over several spots). Other irregular spots may be deleted, though if there is a reasonable number of pixels contained within the spot boundary it should still be usable regardless of shape. The edges of the annulus-shaped spots are often then detected, this helps with establishing the expression level of the spot (refer to Yang *et al.* [60]).
3) Normalization: the intensity of the spots represents the abundance of mRNA being expressed. Normalization is a process that removes any non-biological biases present, for example spatial or dye biases to allow the comparison of spots (genes) both within and between arrays. For most microarrays the majority of genes are not differentially expressed and normalisation approaches such as intensity dependent robust local regression typically perform quite well. Control spots may be used if this is not the case. For more details on different normalisation approaches refer to Smyth and Speed [61].
4) Identification of differentially expressed gene sets: Analysis methods for to identify differentially expressed genes are actively being developed. Linear modelling and empirical Bayes approaches are used to rank genes taking into account multiple testing issues, clustering and discrimination (unsupervised and supervised learning) techniques can be used to distinguish between different classes of treatment or diseases, and time series analysis can be performed to identify differences in gene regulation between samples [62]–[64].

One interesting technique for processing of microarray data is to use a CNNUM (cellular neural network universal machine) [65]. The main components of this electrical hardware are:

1) an array of analog processors, each one connected to all the surrounding processors,
2) a means of storing locally the intermediate computation results for each pixel, and
3) stored, programmable parameters.

The CNNUM is then programmed to implement the following steps:

1) To clear the image from the background noise, a sequence of thresholding and diffusion templates (sets of weights) are applied. This has the effect of quickly and accurately removing the background, even if the background luminosity varies across the microarray.
2) A set of operations are then applied, which first determine the grid in which the spots should lie, and then deletes those spots not in correct positions.
3) Four operations which remove small spots in any of the four directions (up, down, left, right) are then applied to remove those small irregular shaped spots which are too small to be used accurately.
4) Another four operations which remove all unusable large irregular shaped spots are applied that operate similarly to those that remove the small irregular shaped spots.
5) A set of threshold operations are then performed which classify the remaining well-defined spots into a set of expression levels.

As this can be obtained quickly with a high degree of accuracy in a CNNUM chip laid directly on top of a gene chip, it should be possible to build cheaper and faster microarray technologies for real-time analysis.

C. *Time series analysis of gene expression data*

Of interest to geneticists is not only what happens in the expression levels of two different samples at a fixed point in time, but how the expression levels vary over a number of different points in time. A number of different signal processing techniques have been developed to analyze such data, as well as “gene clusters” (sets of related genes) in microarrays. An overview of gene clustering algorithms can be found in Moreau *et al.* [66], and below we discuss some time series approaches.

One approach taken to analysis of time series microarray data, as well as other microarray data, is to take a standard statistical approach to determining factors affecting the output [62]. Bayesian network models have also been used to analyze time series microarray data [63]. With a Bayesian network model, one can efficiently analyze the relationships between the expression levels at different points in time, and between different genes. All the types of analysis that are used for microarray time series data have the property that they can make accurate predictions about gene expression levels based on models with a limited amount of input data. The input data is limited due to the time and cost involved in preparing the microarray data.

One method for analysis of gene expression time series data is that developed by Bar-Jospeh *et al.* [64]. Their method uses mathematical techniques similar to those developed by James and Hastie [67], however it deals properly with gene clusters – groups of related genes, which have correlated expression levels in a microarray analysis. The method fits curves to the limited number of data points available, which are limited due to the cost and time involved in preparing microarrays, and takes into account underlying biological processes and variability. The main steps of the method are outlined in Appendix D. The result of applying the above spline fitting and warping algorithms to gene expression levels in yeast are shown in Figure 12.

D. *Future directions*

As microarray technology evolves, new applications, cheaper platforms, specialised microarrays, and increasing complex experimental designs are likely to be developed [14], [65]. Whilst existing techniques are still likely to be effective (such as the Bayesian network method and spline curve method for analysis of time series data), the generation of new or larger volumes of data will require the development of more sensitive and robust techniques to identify the biological information signal present amongst the noise.

As microarray technology becomes faster and cheaper [14], [65], the analysis of time series microarray data will become more commonplace. The existing techniques used (especially

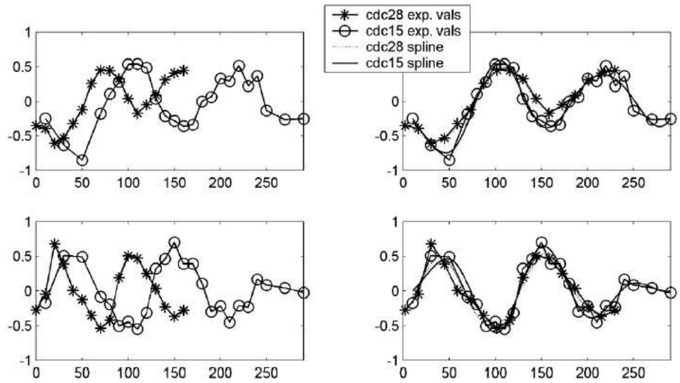


Fig. 12. These graphs (from [64]) show the result of fitting spline curves to sets of time data (the *cdc28* and *cdc15* gene expression levels) in the two left graphs, and the result of applying the warping procedure is shown in the graphs on the right.

the Bayesian network method and spline curve method) will still be effective, however other techniques that work with a larger set of data will be able to be used, since it will be easier to generate larger sets of data. As more time series data is analyzed, it may be possible to build better models for clustering and for predicting the time responses of expression levels in response to a number of factors.

VI. Conclusions

Vast amounts of data are generated by a wide variety of techniques in genetics. Signal processing methods, which have already made a great impact on a number of other areas, are part of a revolution in genetics as they are able to quickly and effectively process the large amount of data. In particular, signal procesing techniques can be used to rapidly process microarray data, making microarrays a much more powerful tool for genetic testing, drug development, and more.

Image processing techniques will certainly aid, and maybe in some areas replace, human analysis of the images such as Figure 7 and Figure 8, which are generated in the study of genetics and biology in general.

Sequence analysis is an exponentially growing industry, as we explore more of the organisms we share our environment with, even the environment inside and on our own bodies. These sequences allow us to produce more effective drugs, better foods, biological solutions to pollution, and to gain valuable insights into the functioning of our own bodies.

Signal processing techniques show promise in being able to complement current techniques in analysis of genetic sequences. The genomes of over ten plants and animals and eighty bacteria are now available, and much of the data would benefit from further exploration. As advances in aerospace technology allowed us to reach out to the stars, so advances in genetic processing allow us to reach out to our own destinies.

Appendix A: Fourier transforms

Fourier transforms (Sections II and III) are used in a wide range applications such as voice prints for evidence in criminal cases, compressing images, and removing noise from music. The discrete Fourier transform is given by

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-2\pi jnk/N}, \tag{9}$$

where $x(n)$ is the sequence of data ($n = 0, \ldots, N-1$), $j = \sqrt{-1}$, k is the discrete frequency, and $X(k)$ is the discrete Fourier transform at frequency k. For DNA sequences, we must transform the DNA sequence $s(n)$ into a numerical sequence $x(n)$, or in some cases several numerical sequences $x_i(n)$. One such transformation is that used by Silverman and Linsker [68]. To a sequence of bases, denoted by $\mathbf{s} = s(1)s(2)\ldots s(N)$, a vector is assigned to each base $s(i)$ as per Eq. 10.

$$\mathbf{x}(i) = \begin{cases} (1,0,0), & s(i) = A, \\ (-1/3, 0, 2\sqrt{2}/3), & s(i) = C, \\ (-1/3, -\sqrt{6}/3, -\sqrt{2}/3), & s(i) = G, \\ (-1/3, \sqrt{6}/3, -\sqrt{2}/3), & s(i) = T. \end{cases} \tag{10}$$

So for example the sequence ATG is represented by the sequence of vectors $(1,0,0), (-1/3, \sqrt(6)/3, -\sqrt(2)/3)$, $(-1/3, -\sqrt(6)/3, -\sqrt(2)/3)$. We then compute the power spectrum

$$P(f) = \sum_{c=1}^{3}\left|\frac{1}{N}\sum_{i=1}^{N} x(i)_c e^{-j2\pi if}\right|^2, \tag{11}$$

where $x(i)_c$ is the c-th component of $\mathbf{x}(i)$, and $j = \sqrt{-1}$. N is the length of the sequence (number of bases). A simpler method is to use indicator functions

$$x(i) = \begin{cases} 1, s(i) = \alpha, \\ 0, \text{otherwise}, \end{cases} \tag{12}$$

for some $\alpha \in \{A, T, C, G\}$ [69]. The power spectra of these two methods are related through Eq. 13 [70],

$$|Y(k)|^2 = \begin{cases} \dfrac{N}{N-1}|X(k)|^2, k \neq 0, \\ \dfrac{N}{N-1}|X(k)|^2 - \dfrac{c}{N-1}, \end{cases} \tag{13}$$

where N is the length of the sequences, c is a constant that varies with N, $X(k)$ is the Fourier transform of the indicator sequence, $Y(k)$ is the average of the Fourier transforms of the sequences of components of the vector sequence as given in Eq. 11.

Appendix B: Hidden Markov models

A Markov model (Section III.A) of order k has a set of states S, with the probability of being in state $s \in S$ being dependent only on the previous k states. So for a discrete time sequence $s_1, \ldots, s_n$,

$$\begin{aligned} &P\left(s_n = i_n \middle| s_{n-1} = i_{n-1}, \ldots, s_1 = i_{n-1}\right) = \\ &\qquad P\left(s_n = i_n | s_{n-1} = i_{n-1}, s_{n-k} = i_{n-k}\right), \end{aligned} \tag{14}$$

except where $k = 0$ and the probability does not depend on the previous states. In a hidden Markov model, the states are unknown and must be inferred from the data. We find a model that maximizes the log likelihood (and thus the likelihood),

$$\log P\left(x|\theta\right) = \sum_{y} P\left(x, y|\theta\right), \tag{15}$$

for x the observed sequence, the y's are the possible sequences, and θ is the set of observed parameters. Then assume there exists a model θ^{t}, and we wish to see if there is a better model θ^{t+1}. Using Bayes' theorem, we can rewrite $\log P\left(x|\theta\right)$ as

$$\log P\left(x|\theta\right) = \log P\left(x, y|\theta\right) - \log P\left(y|x, \theta\right). \tag{16}$$

Using results from information theory, one can show that

$$\log P\left(x|\theta\right) - \log P\left(x|\theta^{t}\right) \geq Q\left(\theta|\theta^{t}\right) - Q\left(\theta^{t}|\theta^{t}\right), \tag{17}$$

where

$$Q\left(\theta|\theta^{t}\right) = \sum_{y} P\left(y|x, \theta^{t}\right) \log P\left(x, y|\theta\right) \tag{18}$$

Setting

$$\theta^{t+1} = \arg\max_{\theta} Q\left(\theta|\theta^{t}\right), \tag{19}$$

will always make the difference positive, and thus the log likelihood of the new model will be greater than the old one, unless $\theta^{t+1} = \theta^{t}$ in which case it stays the same. The above method is the expectation maximization algorithm, and forms the basis of the Baum-Welch algorithm used in hidden Markov model construction [11].

Appendix C: Mutual information

The mutual information function, introduced in Section III.B, for symbols at distance d apart is given in Eq. 20,

$$M(d) = \sum_{\alpha \in \mathcal{A}} \sum_{\beta \in \mathcal{A}} P_{\alpha\beta}(d) \log_2 \frac{P_{\alpha\beta}(d)}{P_{\alpha} P_{\beta}}, \tag{20}$$

for symbols $\alpha, \beta \in \mathcal{A}$ (in the case of DNA, $\mathcal{A} = \{\mathrm{A}, \mathrm{T}, \mathrm{C}, \mathrm{G}\}$). $P_{\alpha\beta}(d)$ is the probability that symbols α and β are found a distance d apart. This is related to the correlation function in Eq. 21 [28]:

$$\Gamma(d) = \sum_{\alpha \in \mathcal{A}} \sum_{\beta \in \mathcal{A}} a_{\alpha} a_{\beta} P_{\alpha\beta}(d) - \left(\sum_{\alpha \in \mathcal{A}} a_{\alpha} P_{\alpha}\right)^{2}, \tag{21}$$

where a_{α} and a_{β} are numerical representations of symbols α and β.

Appendix D: Spline curves

The use of spline curves was introduced in Section V.C. With a spline curve, one approximates a curve using a set of basic functions (often polynomials) that are fitted to the function at a set of points where the function used to approximate the curve can change, but must meet certain specifications (often ones designed to make the spline curve look smooth), and conditions are also specified on the ends of the spline curve. The main steps of the method used by Bar-Joseph *et al.* [64] are:

1) A spline curve model is developed which takes into account the gene cluster information, and fits a curve to the set of data points.
2) If the gene cluster information is not already available, an EM (Expectation and Maximization) algorithm is used to give estimates of the gene cluster information.
3) The curves are then scaled on the time axis, so that different realizations of biological processes can be compared.

In step one, we develop a spline curve using the model

$$Y_i(t) = s(t)(\mu_j + \gamma_i) + \epsilon_i, \tag{22}$$

where $Y_i(t)$ is the observed expression level for gene i at time t, $s(t)$ is a vector containing spline functions, μ_j is the average value of the spline coefficients for genes in cluster (or class) j, γ_i is the gene specific coefficients for gene i, and ϵ_i is a random noise term. If the μ_j or clusters are unknown, they are estimated using the following algorithm (note that MAP stands for Maximum *A Posteriori*):

```
TimeFit($Y, S, c, n$)
  For all classes $j$ {
    choose a random gene $i$
    initialize class center with a random gene
    calculate an initial value of $\mu_j$
  }
  Initialize the other variables
  Repeat until the variables converge {
    E step:
      for all genes $i$ and classes $j$
      compute the conditional probability $p(j|i)$
    M step:
      for all genes $i$ and classes j
        find the MAP estimate of $\gamma_{i,j}$
      Maximize the other variables with respect to $p(j, i)$
      for all classes $j$, $p_j \leftarrow \frac{1}{n}\sum_{i=1}^{n} p(j|i)$
  }
}
```

The spline curves are then aligned using the following method. First denote a reference spline curve (that is, the one we are aligning to) as $g_i^{(1)}(s)$, where $s_{\min} \leq s \leq s_{\max}$, $s_{\min}$ and $s_{\max}$ are the start and end times. The splines to be aligned are denoted $g_i^{(2)}(t)$ for $t_{\min} \leq t \leq t_{\max}$. Then define a mapping for the time as $T(s) = t = (s - b)/a$. The alignment error e_i^2 for each gene is

$$e_i^2 = \frac{\int_{\alpha}^{\beta} \left[g_i^{(2)}(T(s)) - g_i^{(1)}(s)\right]^2}{\beta - \alpha}. \tag{23}$$

The error for a set of genes S of size n is then

$$E_S = \sum_{i=1}^{n} w_i e_i^2, \tag{24}$$

where $w_i = E_S/n$. Minimising E_S numerically then gives the alignment factors α and β.

REFERENCES

[1] P. Clerc and P. Avner, “Role of the region 3’ to Xist exon 6 in the counting process of X-chromosome inactivation,” *Nature Genetics*, vol. 19, no. 3, pp. 249–253, 1998.

[2] J. S. Mattick, “Non-coding RNAs: the architects of eukaryotic complexity,” *EMBO Reports*, vol. 2, no. 11, pp. 986–991, 2001.

[3] D. C. Boyd, A. Pombo, and S. Murphy, “Interaction of proteins with promoter elements the human U2 snRNA genes *in vivo*,” *Gene*, vol. 315, pp. 103–112, 2003.

[4] G. P. Harmer, D. Abbott, P. G. Taylor, and J. M. R. Parrondo, “Brownian ratchets and Parrondo’s games,” *Chaos*, vol. 11, no. 3, pp. 705–714, 2001.

[5] B. Ewing, L. Hillier, M. C. Wendl, and P. Green, “Base-calling of automated sequencer traces using Phred. I. Accuracy assessment,” *Genome Research*, vol. 8, no. 3, pp. 175–185, 1998.

[6] B. Ewing and P. Green, “Base-calling of automated sequencer traces using Phred. II. Error probabilities,” *Genome Research*, vol. 8, no. 3, pp. 186–194, 1998.

[7] J. P. Fitch and B. Sokhansanj, “Genomic engineering: moving beyond DNA sequence to function,” *Proc. IEEE*, vol. 88, no. 12, pp. 1949–1971, 2000.

[8] D. Anastassiou, “Genomic signal processing,” *IEEE Signal Processing Magazine*, vol. 18, pp. 8–20, 2001.

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

[10] Z. Yu, V. Anh, and K. Lau, “Measure representation and multifractal analysis of complete genomes,” *Physical Review E*, vol. 64, no. 3, 2001, 031903.

[11] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[12] A. V. Lukashin and M. Borodovsky, “GeneMark.hmm: new solutions for gene finding,” *Nucleic Acids Research*, vol. 26, no. 4, pp. 1107–1115, 1998.

[13] X.-Y. Zhang, F. Chen, Y.-T. Zhang, S. C. Agner, M. Akay, Z.-H. Lu, M. M. Y. Waye, and S.-W. Tsui, “Signal processing techniques in genomic engineering,” *Proc. IEEE*, vol. 90, no. 12, pp. 1822–1833, 2002.

[14] C. H. Mastrangelo, M. A. Burns, and D. T. Burke, “Microfabricated devices for genetic diagnostics,” *Proc. IEEE*, vol. 86, no. 8, pp. 1769–1787, 1998.

[15] D. Brady, M. Kocic, A. W. Miller, and B. L. Karger, “A maximum-likelihood base caller for dna sequencing,” *IEEE Trans. Biomedical Engineering*, vol. 47, no. 9, pp. 1271–1280, 2000.

[16] N. M. Haan and S. J. Godsill, “Bayesian models for DNA sequencing,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2002, pp. 4020–4023.

[17] M. Pop, S. Salzberg, and M. Shumway, “Genome sequence assembly: algorithms and issues,” *Computer*, vol. 35, no. 7, pp. 47–54, 2002.

[18] C. Bystroff, V. Thorsson, and D. Baker, “HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins,” *Journal of Molecular Biology*, vol. 301, pp. 173–190, 2000.

[19] B. Rost, “Better 1D predictions by experts with machines,” *Proteins: Structure, Function, and Genetics*, vol. Suppl. 1, pp. 192–197, 1997.

[20] S. R. Eddy, “HMMER: Profile hidden Markov models for biological sequence analysis,” 2002, http://hmmer.wustl.edu/.

[21] T. Hayashi, K. Makino, M. Ohnishi, K. Kurokawa, K. Ishii, K. Yokoyama, *et al.*, “Complete genome sequence of enterohemorrhagic *escherichia coli* O157:H7 and genomic comparison with a laboratory strain K-12,” *DNA Research*, vol. 8, pp. 11–22, 2001.

[22] R. Welch, V. Burland, G. Plunkett, *et al.*, “Extensive mosaic structure revealed by the complete genome sequence of uropathogenic *escherichia coli*,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 99, no. 26, pp. 17 020–17 024, 2002.

[23] K. Putegowda, W. Worek, N. Pappas, A. Dandapani, P. Athanas, and A. Dickerman, “A run-time reconfigurable system for gene-sequence searching,” in *Proc. 16th International Conference on VLSI Design*. IEEE Computer Society, Jan. 2003, pp. 561–566.

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

[25] P. P. Vaidyanathan and B.-J. Yoon, “Gene and exon prediction using allpass-based filters,” in *Proc. 36th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2002.

[26] M. Kuroda, T. Ohta, I. Uchiyama, T. Baba, H. Yuzawa, *et al.*, “Whole genome sequencing of meticillin-resistant staphylococcus aureus,” *Lancet*, vol. 357, no. 9264, pp. 1225–1240, 2001.

[27] International Human Genome Sequencing Consortium, “Initial sequencing and analysis of the human genome,” *Nature*, vol. 409, pp. 860–921, 2001.

[28] W. Li, “Mutual information functions versus correlation functions,” *Journal of Statistical Physics*, vol. 60, pp. 823–837, 1990.

[29] P. Bernaolo-Galván, P. Carpena, R. Román-Roldán, and J. L. Oliver, “Study of statistical correlations in DNA sequences,” *Gene*, vol. 300, pp. 105–115, 2002.

[30] T. Higuchi, “Approach to an irregular time series on the basis of the fractal theory,” *Physica D*, vol. 31, pp. 277–283, 1988.

[31] D. Holste, I. Grosse, and H. Herzel, “Statistical analysis of the DNA sequence of human chromosome 22,” *Physical Review E*, vol. 64, no. 4, 2001, 041917.

[32] D. Holste, I. Grosse, S. Beirer, P. Schieg, and H. Herzel, “Repeats and correlations in human DNA sequences,” *Physical Review E*, vol. 67, no. 6, 2003, 061913.

[33] W. Li, “The study of correlation structures of DNA sequences: a critical review,” *Computers and Chemistry*, vol. 21, no. 4, pp. 257–271, 1997.

[34] ——, “Generating non-trivial long-range correlations and 1/f spectra by replication and mutation,” *International Journal of Bifurcation and Chaos*, vol. 2, no. 1, pp. 137–154, 1992.

[35] C.-K. Peng, S. V. Buldyrev, A. L. Goldberger, S. Havlin, F. Sciortino, M. Simon, and H. E. Stanley, “Long-range correlations in nucleotide sequences,” *Nature*, vol. 356, pp. 168–170, 1992.

[36] A. C. Frank and J. R. Lobry, “Oriloc: prediction of replication boundaries in unannotated bacterial chromosomes,” *Bioinformatics*, vol. 16, no. 6, pp. 560–561, 2000.

[37] P. D. Cristea, “Large scale features in DNA genomic signals,” *Signal Processing*, vol. 83, pp. 871–888, 2003.

[38] N. Chomsky, “Three models for the description of language,” *IRE Transactions on Information Theory*, vol. 2, pp. 113–124, 1956.

[39] ——, “On certain formal properties of grammars,” *Information and Control*, vol. 2, pp. 137–167, 1959.

[40] S. Dong and D. Searls, “Gene structure prediction by linguistic methods,” *Genomics*, vol. 23, no. 3, pp. 540–551, 1994.

[41] S. R. Eddy and R. Durbin, “RNA sequence analysis using covariance models,” *Nucleic Acids Research*, vol. 22, pp. 2079–2088, 1994.

[42] R. N. Mantegna, S. V. Buldyrev, A. L. Goldberger, S. Havlin, C.-K. Peng, M. Simons, and H. E. Stanley, “Linguistic features of non-coding DNA sequences,” *Physical review letters*, vol. 73, pp. 3169–3172, 1994.

[43] ——, “Systematic analysis of coding and noncoding DNA sequences using methods of statistical linguistics,” *Physical Review E*, vol. 52, pp. 2939–2950, 1995.

[44] ——, “Reply to comments on linguistic features of non-coding DNA sequences,” *Physical Review Letters*, vol. 76, pp. 1979–1981, 1996.

[45] S. Small, A. Blair, and M. Levine, “Regulation of even-skipped stripe 2 in the *Drosophila* embryo,” *The EMBO Journal*, vol. 11, no. 11, pp. 4047–4057, 1992.

[46] P. Bucher and A. Bairoch, “A generalized profile syntax for biomolecular sequences motifs and its function in automatic sequence interpretation,” in *ISMB-94; Proceedings 2nd International Conference on Intelligent Systems for Molecular Biology*, R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, Eds., Aug. 1994, pp. 53–61.

[47] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. Sigrist, K. Hofmann, and A. Bairoch, “The PROSITE database, its status in 2002,” *Nucleic Acids Research*, vol. 30, pp. 235–238, 2002.

[48] M. Ortuño, P. Carpena, P. Bernaola-Galván, E. Muñoz, and A. M. Somoza, “Keyword detection in natural languages and DNA,” *Europhysics Letters*, vol. 57, no. 5, pp. 759–764, 2002.

[49] M. J. Berryman, A. Allison, and D. Abbott, “Statistical techniques for text classification based on word recurrence intervals,” *Fluctuations and Noise Letters*, vol. 3, no. 1, pp. L1–L10, 2003.

[50] B. M. E. Moret, J. Tang, L.-S. Wang, and T. Warnow, “Steps toward accurate reconstructions of phylogenies from gene-order data,” *Journal of Computer and Systems Sciences*, vol. 65, no. 3, pp. 508–525, 2002.

[51] R. Mutihac, A. Cicuttin, and R. C. Mutihac, “Entropic approach to information coding in DNA molecules,” *Materials Science and Engineering C*, vol. 18, pp. 51–60, 2001.

[52] M. J. Berryman, A. Allison, and D. Abbott, “Stochastic evolution and multifractal classification of prokaryotes,” *Proc. of the SPIE: Fluctuations and Noise in Biological, Biophysical, and Biomedical Systems*, vol. 5110, pp. 192–200, June 2003.

[53] R. Kumar, R.-A. Wang, A. Mazumda, A. H. Talukder, M. Mandal, *et al.*, "A naturally occurring MTA1 variant sequesters oestrogen receptor-alpha in the cytoplasm," *Nature*, vol. 418, pp. 654–657, 2002.

[54] P. Eerdewegh, D. Little, J. Dupuis, R. G. Mastro, *et al.*, "Association of the ADAM33 gene with asthma and bronchial hyperresponsiveness," *Nature*, vol. 418, pp. 426–430, 2002.

[55] J. Russ, *The Image Processing Handbook*. CRC Press, Inc., 1995.

[56] L. Roberts, *Optical and Electro-optical Information Processing*. MIT Press, 1965, ch. 9, pp. 159–197.

[57] "Nature genetics chipping forecast," pp. Supplement 1–60. [Online]. Available: http://www.nature.com/cgi-taf/DynaPage.taf?file=/ng/journal/v21/n1s/index.html

[58] "Nature genetics chipping forecast ii," pp. Supplement 461–552. [Online]. Available: http://www.nature.com/cgi-taf/DynaPage.taf?file=/ng/journal/v32/n4s/index.html

[59] J. L. de Risi, 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. 680–687, 1997.

[60] 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, pp. 108–136, 2002.

[61] G. K. Smyth and T. P. Speed, *Normalization of cDNA microarray data*, 2003.

[62] L. P. Zhao, R. Prentice, and L. Breeden, "Statistical modeling of large microarray data sets to identify stimulus-response profiles," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 98, no. 10, pp. 5631–5636, 2001.

[63] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Proc. Fourth Annual International Conference on Computational Molecular Biology*, pp. 127–135, Apr. 2000.

[64] Z. Bar-Joseph, G. Gerber, D. Gifford, and T. Jaakkola, "A new approach to analyzing gene expression time series data," in *Proc. Sixth Annual International Conference on Research in Computational Molecular Biology*, Apr. 2002, pp. 39–48.

[65] P. Arena, L. Fortuna, and L. Occhipinti, "A CNN algorithm for real time analysis of DNA microarrays," *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, vol. 49, no. 3, pp. 335–340, 2002.

[66] 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, Nov. 2002.

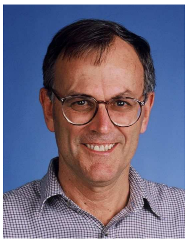
[67] G. James and T. Hastie, "Functional linear discriminant analysis for irregularly sampled curves," *Journal of the Royal Statistical Society Series B*, vol. 63, pp. 533–550, 2001.

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

[69] S. Tavaré and B. W. Giddings, *Mathematical Methods for DNA Sequences*. CRC Press, 1989, pp. 117–131.

[70] E. Coward, "Equivalence of two Fourier methods for biological sequences," *Journal of Mathematical Biology*, vol. 36, pp. 64–70, 1997.

Matthew J. Berryman (S'03) received a B.Sc. in Mathematical and Computer Sciences and a B.E. (Hons.) in Computer Systems from The University of Adelaide, Australia. He is currently a Ph.D. candidate at The University of Adelaide at the Centre for Biomedical Engineering, working on several areas in bioinformatics including signal processing of DNA and analysis of EEG signals. In 2003, he won a Santa Fe Institute CSSS scholarship and was also a visiting scholar at the Center for the Study of Complex Systems at the University of Michigan, Ann Arbor. He is the treasurer of the Electrical and Electronic Engineering Society of Adelaide University, an IEEE affiliated student group.



Douglas A. Gray (M'82) was born in the U.K. in 1946. He holds the Professorial Chair of Sensor Signal Processing at The University of Adelaide, Australia. He is also Deputy Director of the Co-operative Research Centre for Sensor Signal and Information Processing (CSSIP), leading the Sensor Signal Processing Program. He received the B.Sc. degree (1969) in mathematical physics and a Ph.D. (1974) in mathematical physics, in the area of quantum field theory, both from The University of Adelaide, Australia.

From 1973 to 1993, he was with the Australian Defence Science and Technology Organisation (DSTO), Adelaide, applying signal processing to sonar and electronic warfare. From 1977 to 1979, he was a Visiting Scientist at the Royal Aircraft Establishment (RAE), and in 1985, he was a Visiting Fellow at the Australian National University (ANU), Canberra. His research interests are in the application of signal processing to sonar, radar, GPS, electronic warfare, and electricity grids, particularly in adaptive processes, beamforming, signal sorting and classification techniques, data fusion, estimation, and system identification.

Christopher Wilkinson received a B.Sc. (Hons) in experimental physics in 1993 and a Ph.D. in High Energy Astrophysics in 1998 from the University of Adelaide, Australia. He received a commendation for this Ph.D. thesis on the application of high precision timing to the High Resolution Fly's Eye Cosmic Ray detector. From 1998 to 2001 he worked in the Australian Defence Science and Technology Organisation's (DSTO) terminal effects group where he performed experimental and computational studies of explosively generated air and underwater shock effects. From 2001 to 2003 he worked in Professor Terry Speed's Bioinformatics group at the Walter and Eliza Hall Institute for Medical Research (WEHI), Australia on a project funded through the co-operative Research Centre for the discovery of genes for common human diseases to identify genetic regions associated with Multiple Sclerosis. In 2003 he joined the Child Health Research Institute, Australia, to perform microarray and bioinformatics analysis on a study of myeloid cell proliferation, differentiation and leukeamia. He is also a visiting research fellow in the microarray analysis group based in the School of Applied Mathematics at the University of Adelaide.

Andrew Allison (S'90-M'94) graduated from The University of Adelaide with B.Sc in mathematics and BEng.(Hons) in computer systems engineering in 1978 and 1995 respectively. He worked briefly in biochemical laboratories for Barratt Bros. Maltsters and later for the Division of Horticultural Research of the Commonwealth Scientific Industrial Research Organization (CSIRO) where he worked on the mathematical analysis of reaction dynamics of recombination of DNA. He worked for 14 years at Telecom, AOTC and Telstra in many areas including: telephony, electronic data processing, computer networks and cable television. Since 1996 he has worked as a lecturer in the Department of Electrical and Electronic Engineering at The University of Adelaide. He is a member of SA committee of the IEEE and a member IEAust.

Derek Abbott (M’85-SM’99) was born in South Kensington, London, U.K. He received the B.Sc.(HONS) degree in physics from Loughborough University of Technology, U.K., and the Ph.D. (with commendation) degree in electrical and electronic engineering from The University of Adelaide, Australia. He has led a number of research programs in the imaging arena, ranging from the optical to infrared to millimeter wave to terahertz (T-ray) regimes. From 1977 to 1986, he worked at the GEC Hirst Research Centre, London, U.K., in the area of visible and infrared image sensors. He has worked with nMOS, CMOS, SOS, CCD, GaAs, and vacuum microelectronic technologies and is now also in the MEMS/bio/nanotechnology arena. His expertise spans VLSI design, optoelectronics, device physics, noise, fabrication, and testing. On emigrating to Australia, he worked for Austek Microsystems, Technology Park, South Australia.

Since 1987, he has been with The University of Adelaide, where he is presently an Associate Professor and the Director of the Centre for Biomedical Engineering (CBME). He is also a full professor under an adjunct appointment at Edith Cowan University, Perth, Australia. He is a founder member of the Centre for GaAs VLSI Technology (now CHiPTec), instituted in 1987, and served in an acting Deputy Director role (1989-1996) and was a Deputy Director (1996-1998). He has been consultant to various U.K. and Australian defense and industry organizations. He has appeared on national and international television and radio and has also received scientific reportage in *New Scientist*, *The Sciences*, *Scientific American*, *Nature*, *The New York Times*, and *Sciences et Avenir*. He was the discoverer of the photovoltaic self-biasing internal-gain edge-effect within planar GaAs MESFETs and holds over 300 publications/patents. He has reviewed for IEEE TRANSACTIONS and is he co-authoring a text on noise for Cambridge University Press (CUP).

Prof. Abbott has been an invited speaker at over 80 institutions around the world, including Princeton, NJ; MIT, MA; Santa Fe Institute, NM; Los Alamos National Laboratories, NM; Cambridge, U.K.; Technion, Israel, and EPFL, Lausanne, Switzerland. He won the GEC Bursary (1977) and the Stephen Cole the Elder Prize (1998). He has served as an editor and/or guest editor for a number of journals including IEEE JOURNAL OF SOLID-STATE CIRCUITS, *Chaos* (AIP), *Smart Structures and Materials* (IOP), *Journal of Optics B* (IOP), *Microelectronics Journal* (Elsevier), and *Fluctuation Noise Letters* (World Scientific). He is a fellow of the Institute of Physics (IOP) and has served on a number of IEEE and SPIE conference technical program committees, including the *IEEE APCCS* and the *IEEE GaAs IC Symposium*.