Generalised Fourier Analysis of Human Chromosome Images

Allan G. Hanbury
Department of Physics
University of Cape Town
7701 Rondebosch

May 11, 1999

Abstract

Human chromosome classification or karyotyping is routinely carried out in medical genetics laboratories in order to detect genetic abnormality and damage due to environmental factors, or for diagnosing cancer. In order to enable each human chromsome to be classified into one of twenty-three pairs, staining techniques are used which produce a banding pattern on the chromosomes. This pattern, as well as other features such as length and position of the centromere (a constriction in the chromosome) are used to make the classification. Much research has been done into the automation of this task, although the performance of automated classifiers has not yet matched that of an experienced cytogeneticist.

In this dissertation, a new method of analysing chromosome banding patterns based on generalised Fourier analysis techniques is proposed. This involves defining a set of twenty-four average chromosome profiles, with each average representing one type of chromosome. Using this set of average profiles, a set of orthogonal profiles is constructed. By correlating these orthogonal profiles with chromosome profiles, a set of band-description features, which can be used for classification, is extracted.

This dissertation commences by presenting a brief introduction to chromosome karyotyping and a general introduction to pattern recognition, followed by a discussion of previous chromosome classification methods in which these techniques have been used. Then, the construction of average chromosome profiles and orthogonal profiles is described. Finally, classification using the features calculated using the orthogonal profiles and the features derived from previously defined band descriptors are compared.

Contents

Chapter 1

Overview of Cytogenetics and Computer-Assisted Karyotyping

Human chromosomes are routinely examined in medical genetics laboratories for detecting genetic abnormality, damage due to environmental factors or diagnosing cancer. In order to assist in making diagnoses, chromosomes must first be classified into groups and arranged on a standard diagram to produce a karyotype. This is a very time consuming process which can be automated to some extent by computer, although many years of attempts at solutions to the problem of computerised karyotyping have still not produced algorithms able to perform as well as cytogeneticists or cytogenetic technicians.

This chapter presents an overview of genetics and specifically chromosome karyotyping, and then describes the process of computer-assisted chromosome karyotyping.

1.1 Introduction to genetics

A very brief introduction to human genetics is presented in this section[1]. Every living organism is made up of a large number of *cells*, the majority of which contain *nuclei*. Each *nucleus* contains hundreds of thousands of *genes*, which carry the genetic code for the organism, and are bound together in large groups on bodies known as *chromosomes*. The number of chromosomes in the nucleus is dependent on the organism. The number of chromosomes in a human cell was conclusively observed to be 46 by Tjio and Levan (1956) [80]. For comparison, dogs have 78 chromosomes, the housefly has 12 and the tobacco plant has 48. In humans, the 46 chromosomes are divided into 1 pair of sex chromosomes and 22 pairs of *autosomes* (non-sex chromosomes). The two chromosomes in each autosome pair are identical, while the pair of sex chromosomes consist of two identical X chromosomes in females and different X and Y chromosomes in males.

[1]More detail can be found in books on genetics such as those by Connor and Ferguson-Smith [7], Goodenough [17] and Winchester and Mertens [86].

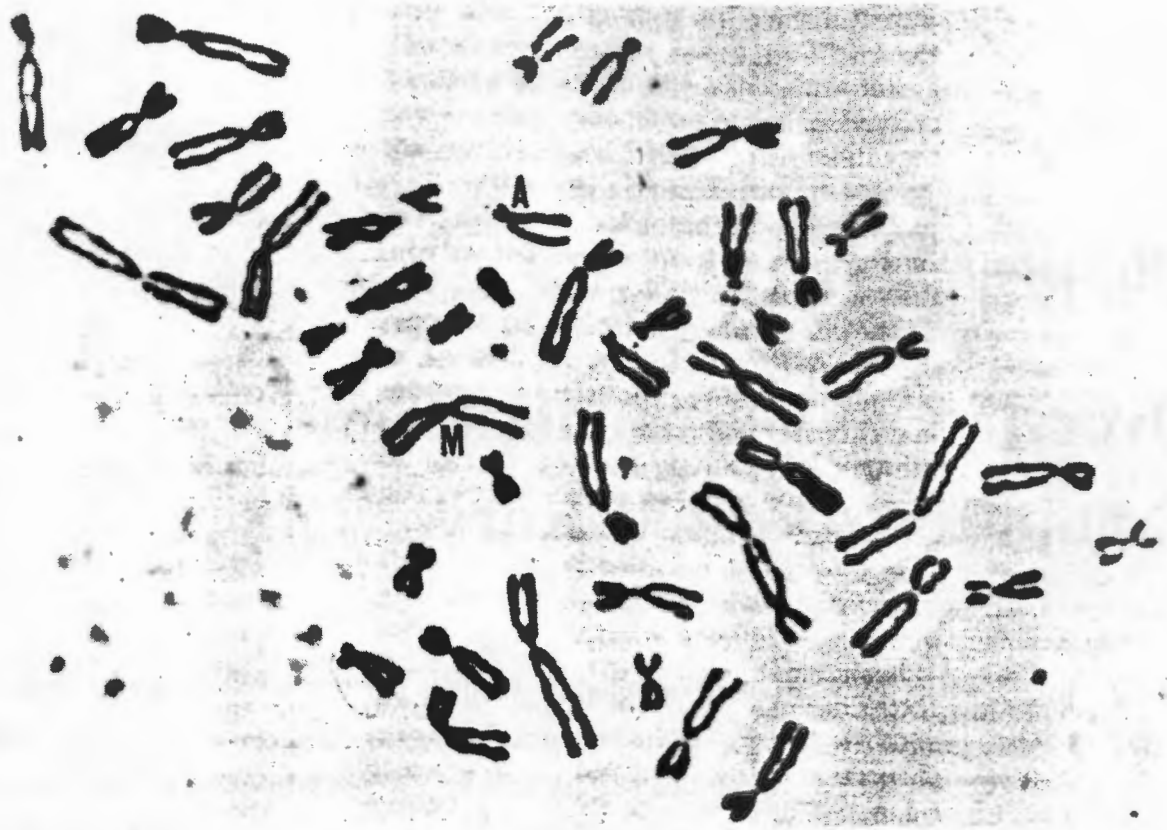


Figure 1.1: An image of homogeneously stained chromosomes of a human male. The two chromatids and the centromere are visible on most of the chromosomes. The chromosome marked "A" is an example of an acrocentric chromosome, and the chromosome marked "M" is an example of a metacentric chromosome (from [17]).

The life cycle of a normal cell consists of alternating periods of growth and division. During the period between cell division, called *interphase*, the chromosomes are very long and slender, and are usually only visible when examined using an electron microscope. During interphase, the genes replicate until each chromosome is double, made up of two *chromatids* held together by a single *centromere*.

During the division stage, called *mitosis*, the chromosomes become progressively shorter and thicker by coiling and folding. At this stage, after the necessary staining, the chromosomes are visible under an optical microscope. The chromosomes have a width of $\sim 1\mu$m and vary in length from $\sim 1\mu$m for the smallest to 15 μm for the largest, depending on the state of contraction of the cell [44]. Figure 1.1 shows a set of homogeneously stained human chromosomes during mitosis. The two chromatids and the centromere (the constriction where the chromatids join) are visible on most of the chromosomes in the image. Cell division continues with the chromatids becoming separated and moving to opposite ends of the cell, after which the cell divides. The cell then enters interphase again.

It can be seen in Figure 1.1 that the chromosomes differ both in size and in the position

of the centromere. It was decided at a meeting in Denver in 1961 to number the pairs of human chromosomes from 1 to 22 according to length, with the two sex chromosomes designated as the twenty-third pair.

The position of the centromere is constant for a given class of chromosome, and a chromosome class may be described as:

1. **Metacentric** - centromere in the middle.

2. **Acrocentric** - centromere close to one end.

3. **Submetacentric** - intermediate position of centromere.

The centromere divides each chromosome into a long and a short arm. The short arm, called the *p arm* is taken to be the top of the chromosome. The longer arm is referred to as the *q arm*. The positions of the p arm and q arm are sometimes referred to as the *polarity* of the chromosome. Due to the composition of chromosomes 13–15 and 21–22, the ends of the p arms sometimes appear as *satellites*, separated from the rest of the chromosome by narrow stalks. The total length of the chromosome divided by the length of the p arm is known as the *length centromeric index*. It is also possible to calculate an *area centromeric index* by dividing the area of the p arm by the total chromosome area, and a *density centromeric index* by dividing the total optical density of the p arm by the total optical density of the chromosome.

Since some of the chromosome pairs can be grouped together based on similarities in length and centromeric indices, it was also decided to divide the chromosomes into seven groups labelled by the letters A to G. Table 1.1 shows the seven groups and the general characteristics of the chromosomes included in each. This division into seven groups is often referred to as the *Denver classification*, with the groups referred to as *Denver classes*.

Due to the similarities in length and centromere position of many types of chromosome, it is not possible to classify a homogeneously stained chromosome (such as those in Figure 1.1) as belonging to a specific pair with any great confidence. Around 1970, new staining techniques were developed that produced characteristic transverse banding patterns along the arms of each chromosome type [26]. The Giemsa stain [76] is a popular method at present. Bands resulting from Giemsa staining are visible using an optical microscope, and are called G-bands to distinguish them from bands produced by other staining techniques (Q-bands and R-bands).

The banding patterns allow for easier differentiation between the twenty-four chromosome classes, and allow a trained person to classify and present the chromosomes in the form of a karyotype with relative ease. Figure 1.2 shows a karyotype of very high quality G-banded chromosomes, and Figure 1.3 presents some ideograms which show the positions of the bands that should be visible on G-banded chromosomes.

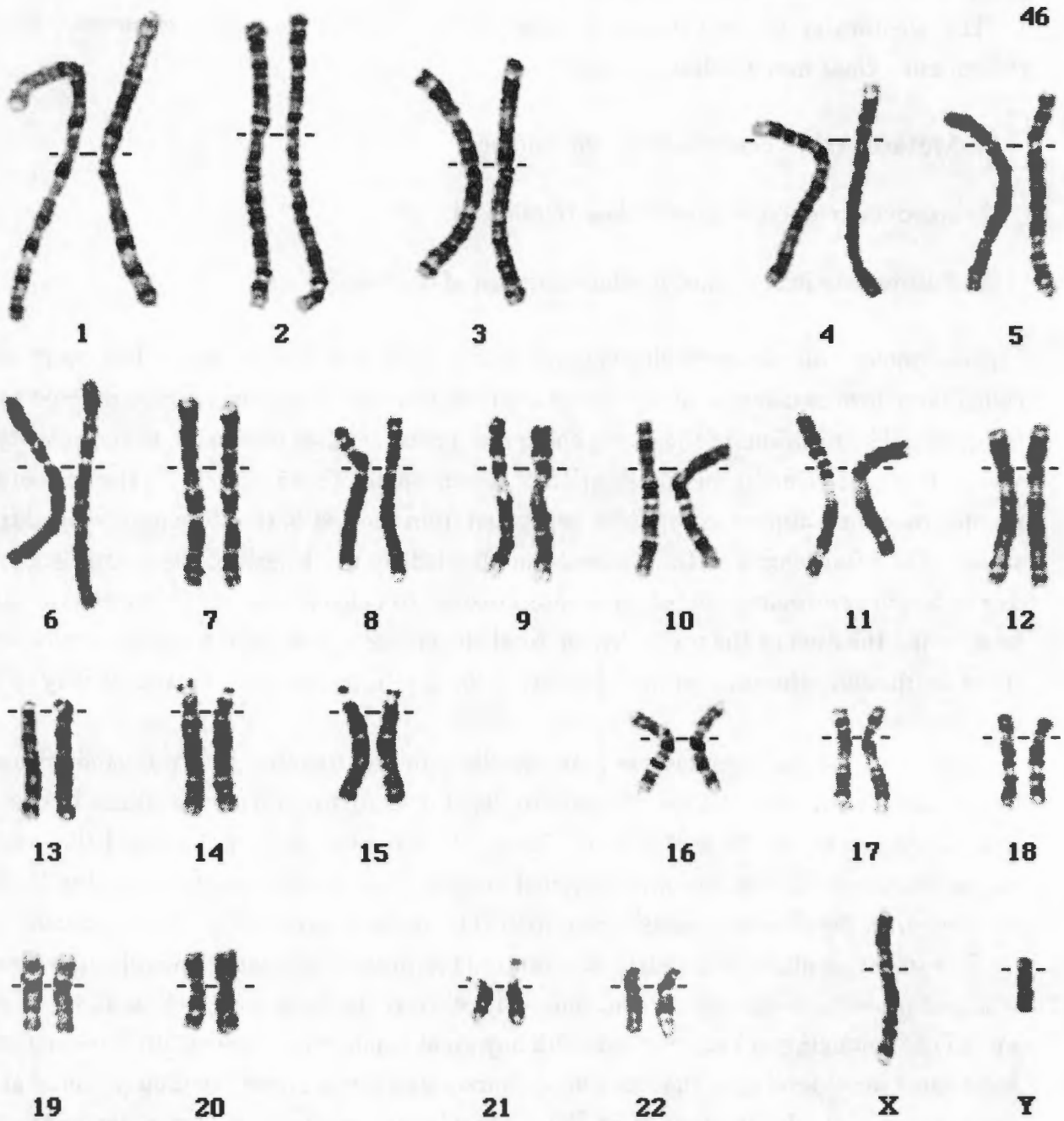


Figure 1.2: Very high quality G-banded chromosomes displaying a total of 550 bands created as a composite of the best chromosomes from five chromosome spreads. It was created by David McDonald of the Laboratory of Pathology of Seattle and obtained from the Primate Cytogenetics Network at http://www.selu.com/~bio/cyto/index.html.

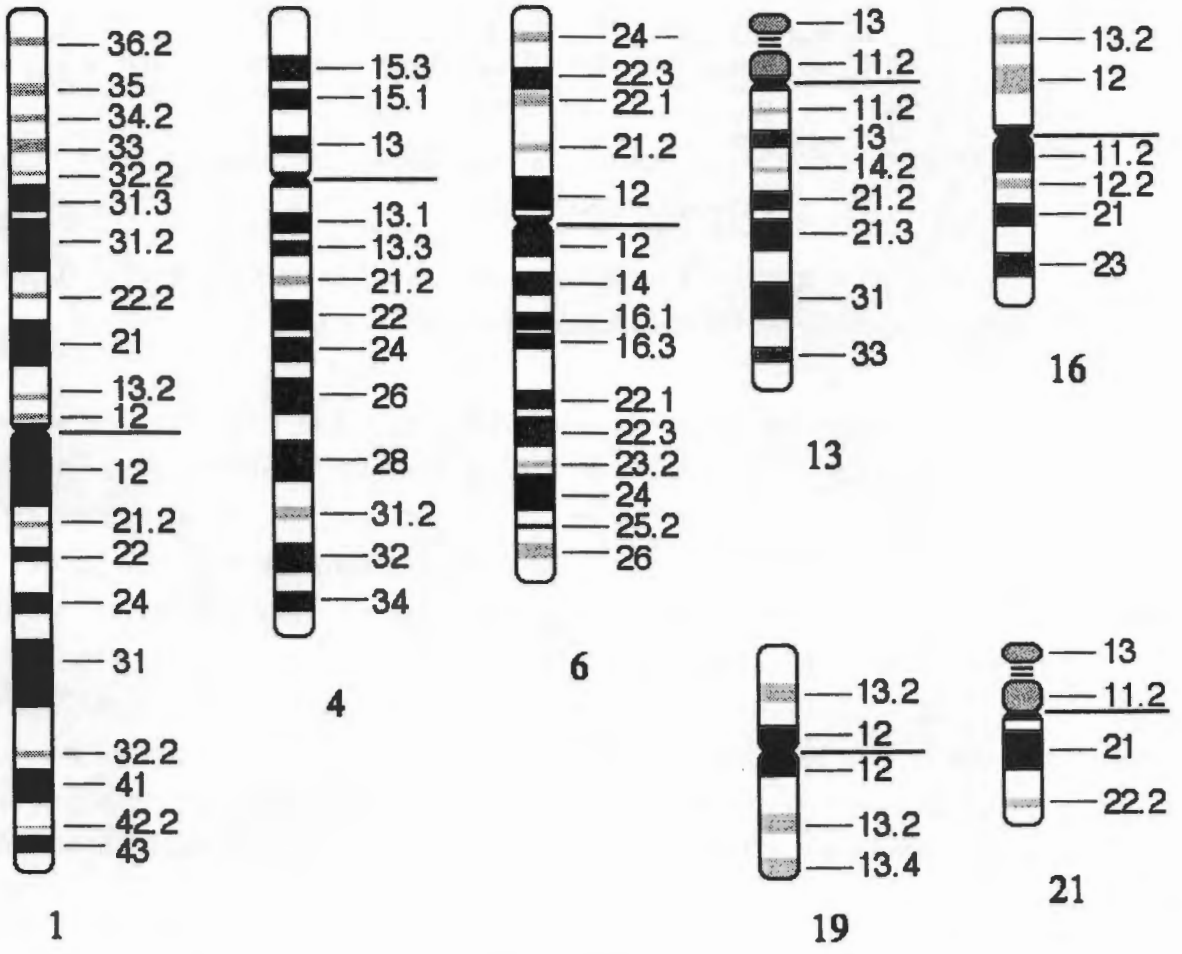


Figure 1.3: Ideograms of seven G-banded chromosomes. The positions of the centromeres are marked by thick black lines. The arms above and below the centromeres are divided into regions, with region 1 being closest to the centromere. Each band in each region is then numbered. For example, 1q24 refers to the 4th band in the second region on the q arm of chromosome 1. The ideograms were created by Tim Knight and obtained from the Primate Cytogenetics Network at http://www.selu.com/~bio/cyto/index.html.

Denver Class	Chromosome Classes	Nature of Chromosome
A	1–3	Very long with approximately metacentric centromeres
B	4–5	Long with submetacentric centromeres
C	6–12 and X	Medium length with submetacentric centromeres
D	13–15	Medium length with acrocentric centromeres, with satellites on short arms
E	16–18	Somewhat short with submetacentric centromeres
F	19–20	Short with metacentric centromeres
G	21–22 and Y	Very short with acrocentric centromeres; 21 and 22 may have satellites

Table 1.1: The division of human chromosomes into seven Denver classes based on similarities in length and centromeric indices (from [86]).

Once the chromosomes have been classified and arranged as a karyotype, it is possible to do further analysis and look for various chromosomal abnormalities. These can be due to an incorrect number of chromosomes present in a cell, or aberrations in chromosome structure. The numerical abnormalities are *trisomies* (a third chromosome for one type) and *monosomies* (a chromosome missing). The best known trisomy is Down's syndrome, which is associated with cells containing three type 21 chromosomes, and an example of a monosomy is Turner's syndrome in females, which is associated with a missing X chromosome. Structural abberations outlined by Habbema [26] include *translocations* where parts of one chromosome have become attached to another chromosome and *deletions* where parts of a chromosome are missing. For example, chronic myeloid leukemia is characterised by a reciprocal translocation between chromosome 22 and chromosome 9.

1.2 Routine karyotyping

The traditional method of karyotyping chromosomes in medical genetics laboratories is outlined below:

1. The chromosome spreads are prepared on microscope slides.

2. Good chromosome spreads suitable for further analysis are located by examining the microscope slides under low magnification.

3. These good chromosome spreads are viewed under high magnification and photographed using a camera mounted on the microscope.

Figure 1.4: Image of a routine chromosome spread before karyotyping (courtesy of the Natal Institute of Immunology).

4. The photographs are developed, and chromosomes are manually cut out of the photograph and pasted onto a sheet of paper to form a karyotype.

The chromosome staining in routine work is usually not done to the same standard as that shown in Figure 1.2. This results in shorter chromosomes with fewer bands visible on the chromosomes. A further complication arises as chromosomes in different spreads contract by different amounts, so the same class of chromosome from different spreads can have different numbers of bands visible. Another problem encountered are overlapping and touching chromosomes. Figure 1.4 shows a typical chromosome spread. Notice the overlapping chromosomes just below the centre of the image. Figure 1.5 shows a karyotype of the chromosomes in Figure 1.4.

1.3 Computer-assisted karyotyping

Computers can be used to simplify steps 2, 3 and 4 above. If the camera mounted on the microscope is replaced with a video camera connected to a computer, then large increases in productivity are possible. The simplest way of using this system is to capture the image and then print it out. Even though the chromosomes still have to be manually cut out

Figure 1.5: The karyotype of the chromosome spread in Figure 1.4 (courtesy of the Natal Institute of Immunology).

and glued onto the karyotype, the necessity to spend time developing the photographs is eliminated. Further gains can be made by installing appropriate software on the computer. The simplest software can be described as "electronic scissors", and allows the operator to cut and paste the chromosomes on the computer display and arrange them as a karyotype which can be printed. More advanced software automates the various steps to some extent and also possibly provides database functions allowing images and details to be stored electronically. Computerised metaphase finding systems are also available.

There has been much research into automating karyotyping, the earliest before the discovery of chromosome banding patterns. One of the earlier karyotyping systems [42] (1966) used an instrument called a FIDAC (Film Input to Digital Automatic Computer) to scan and digitise photomicrographs of chromosomes. The analysis was carried out by an IBM 7094 computer. The authors reported that major limits on speed and the number of points sampled in an image were imposed by the computer, and hoped that increases in computer memory size and speed would increase accuracy and performance.

Good automated karyotyping systems in the 1980's and early 1990's tended to be based on dedicated hardware. One of the most well-known is the Cytoscan [44], which was developed at the UK Medical Research Council Clinical and Population Cytogenetics Unit in Edinburgh, and marketed by Image Recognition Systems, Ltd. It can do metaphase finding and automatic chromosome separation and classification. It uses a Motorola M68000 and VME bus-based multiprocessor configuration. This consists of a master M68000 processor and up to ten slave M68000 processors. The time taken by this system for a full karyotype analysis including operator interaction time is $\sim$2.5 minutes on good quality material. The main disadvantage of these systems was price. According to Lundsteen and Philip [48], the price of karyotyping systems in 1989 was between 50 000 and 200 000 US dollars. The performance of four of these systems is compared by Korthof and Carothers [41].

At this time there were some software-based products available which ran on standard IBM compatible personal computers or Apple Macintoshes[2]. Due to the limitations of the personal computers, many of these products tended to be limited to very simple operations and provided little or no automation of the process. The capabilities of software based products have improved dramatically with the increases in personal computer power. There are presently a few very powerful software karyotyping products on the market. Unfortunately, the best software products are still expensive.

[2]For example, see the description of a semi-automated karyotyping system implemented on an Apple Macintosh II computer by van Vliet, Young and Mayall [83].

Component	Description
Motherboard	Intel Endeavor
Processor	Intel 100 MHz Pentium
Graphics Card	Diamond Stealth 64 Video 3000
Monitor	Philips 20B (20 inch display)
SCSI Controller	Adaptec 2940
Hard Drive	1 GB SCSI
CD-ROM	Hewlett Packard 2040i writable CD-ROM drive
Frame Grabber	Mutech MV-1000 PCI bus frame grabber
Camera	Sony XC77-CE
Printer	Sony UP-930 Thermal Video Printer

Table 1.2: Summary of the hardware used in the implementation of the karyotyping system in use at the Natal Institute of Immunology.

1.4 The karyotyping system implemented at the Natal Institute of Immunology

This section describes the hardware and software of a personal computer based computer-assisted karyotyping system implemented at the Natal Institute of Immunology in 1996. At this stage, there is very little automation in the software, but routines discussed in this dissertation may be added in future. Section 1.4.2 presents an overview of how the software is presently used for karyotyping.

1.4.1 Modern image analysis hardware

Historically, due to the slow data throughput speed of the standard PC AT bus running at 8 MHz, IBM-compatible PC-based image analysis hardware consisted of a computer with two monitors, a standard SVGA monitor connected to the SVGA card, and a video monitor connected to the frame grabber[3]. The image captured by the video camera was stored in memory on the frame grabber and displayed on the video monitor. During image acquisition, the output from the camera could be displayed in real time on the video monitor.

With the invention of the PCI bus, single monitor PC-based image analysis workstations became possible. Images can be transferred at the standard video rate of 25 frames per second (or faster) from the frame grabber to the video card for display on the SVGA monitor, or to system memory. This type of equipment was used for the karyotyping system. The hardware used is summarised in Table 1.2 and diagrammed in Figure 1.6.

[3]The frame grabber is the expansion card which digitises the analogue video signal produced by standard video cameras.

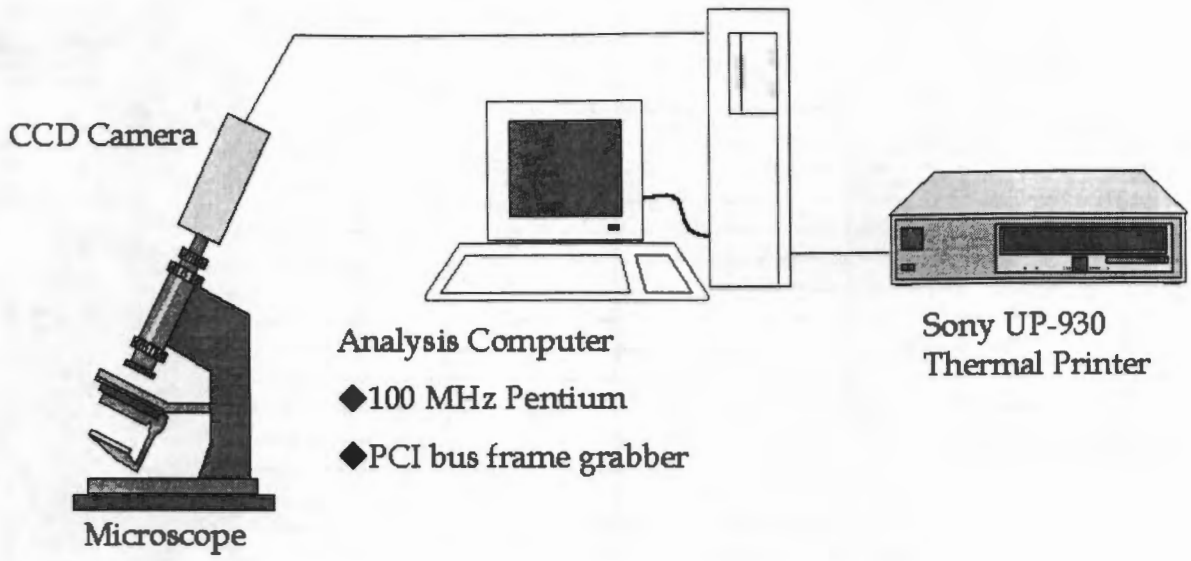


Figure 1.6: Schematic Diagram of the Karyotyping System implemented at the Natal Institute of Immunology.

1.4.2 The software

The software, called "Karyopt", was initially written as a set of macros for Optimas 5.2 and later upgraded to run under Optimas 6.2 [58], an image analysis package originally written and distributed by Optimas Corporation in Seattle, Washington; but now distributed by Media Cybernetics in Silver Spring, Maryland. The image database functions are provided by Optimas Library [57] from the same company. Optimas sends commands to Library using Windows Dynamic Data Exchange (DDE).

Upon starting the Karyopt software, one is presented with the database, Optimas Library, in which all the patient details and images are stored, along with a set of controls which allow one to view the records, add and remove records and search the records (Figure 1.7).

When the "New metaphase" option is chosen, the software prompts for patient details to be entered, and then allows the user to capture an image from the camera. There is also an option to cut and paste chromosomes from separate images for cases in which there are outlying chromosomes which do not appear in the main field of view. When the captured image is satisfactory, the user interactively sets the threshold to separate chromosomes from the background, and the Optimas automatic object marking routine is used to mark the chromosomes, as shown in Figure 1.8. One now has the option of manually correcting faulty separation, for example, separating two chromosomes which have been marked as one object.

Once all the chromosomes have been marked as separate objects, they are automatically sorted by length and displayed as shown in Figure 1.9. The controls on the right of the screen allow one to move chromosomes from the boxes on the right to numbered boxes on

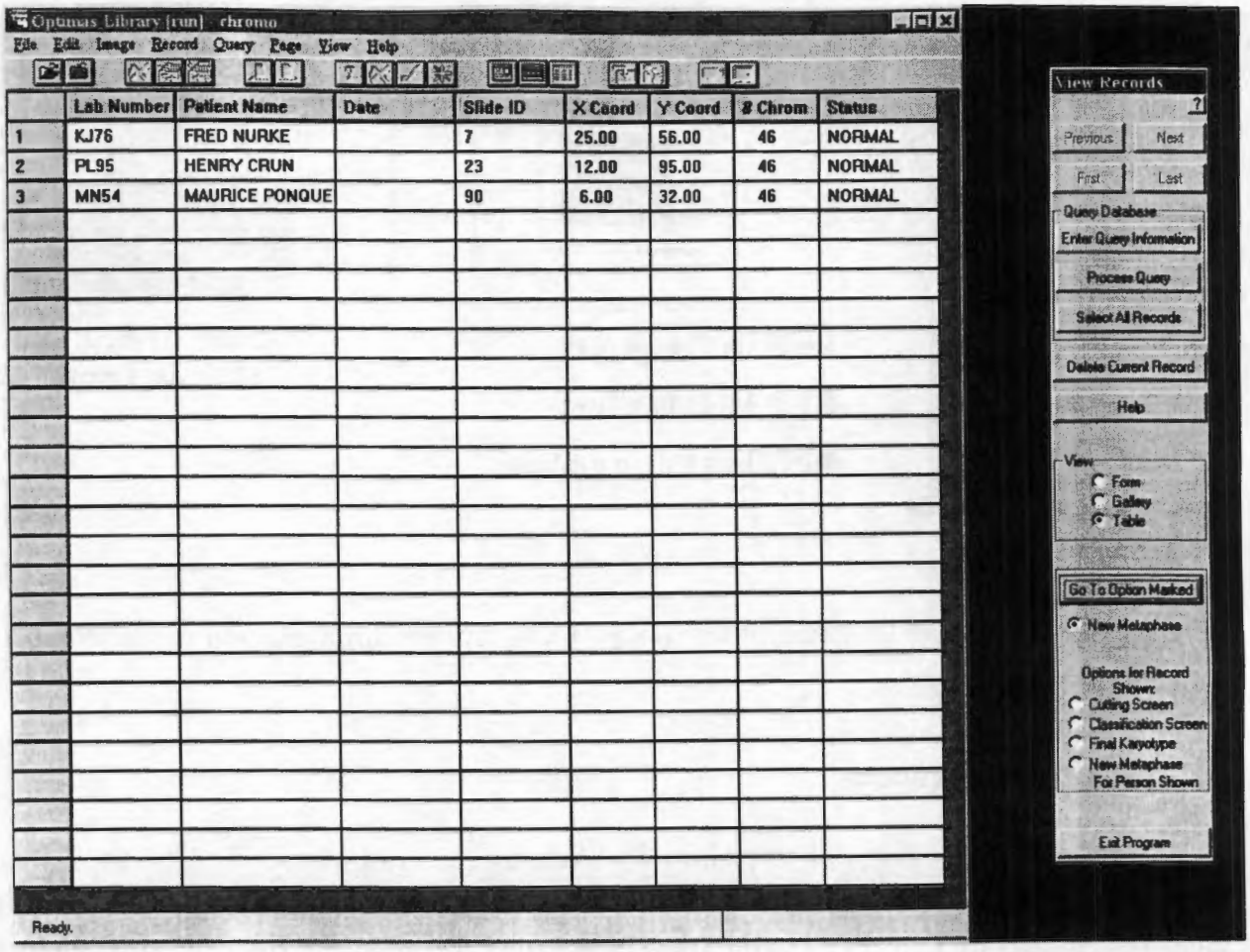


Figure 1.7: The initial Karyopt screen, showing the database and controls.

the left and correct orientation and rotation of the chromosomes. Once all the chromosomes are moved into correctly numbered boxes, the software creates a karyotype similar to the one shown in Figure 1.5. This karyotype is stored in the database, and can be printed. The incorporation of automated classification routines will result in a number of chromosomes being preclassified into numbered boxes, with the user then having the option of correcting the classification.

1.5 Standard data sets

A number of standard data sets, described in Table 1.3, are available for testing classifiers. The chromosomes in the data sets are grouped together into cells. The Copenhagen set is considered to be of very high visual quality as the measurements were carefully made on selected cells of high quality. The Edinburgh, Philadelphia, CDAR and Cpr datasets were digitised from routine material, with no exclusion of, or correction to bent and overlapped chromosomes. The length statistics for the Copenhagen, Edinburgh, Philadelphia and Cpr data sets are given in Appendix B.

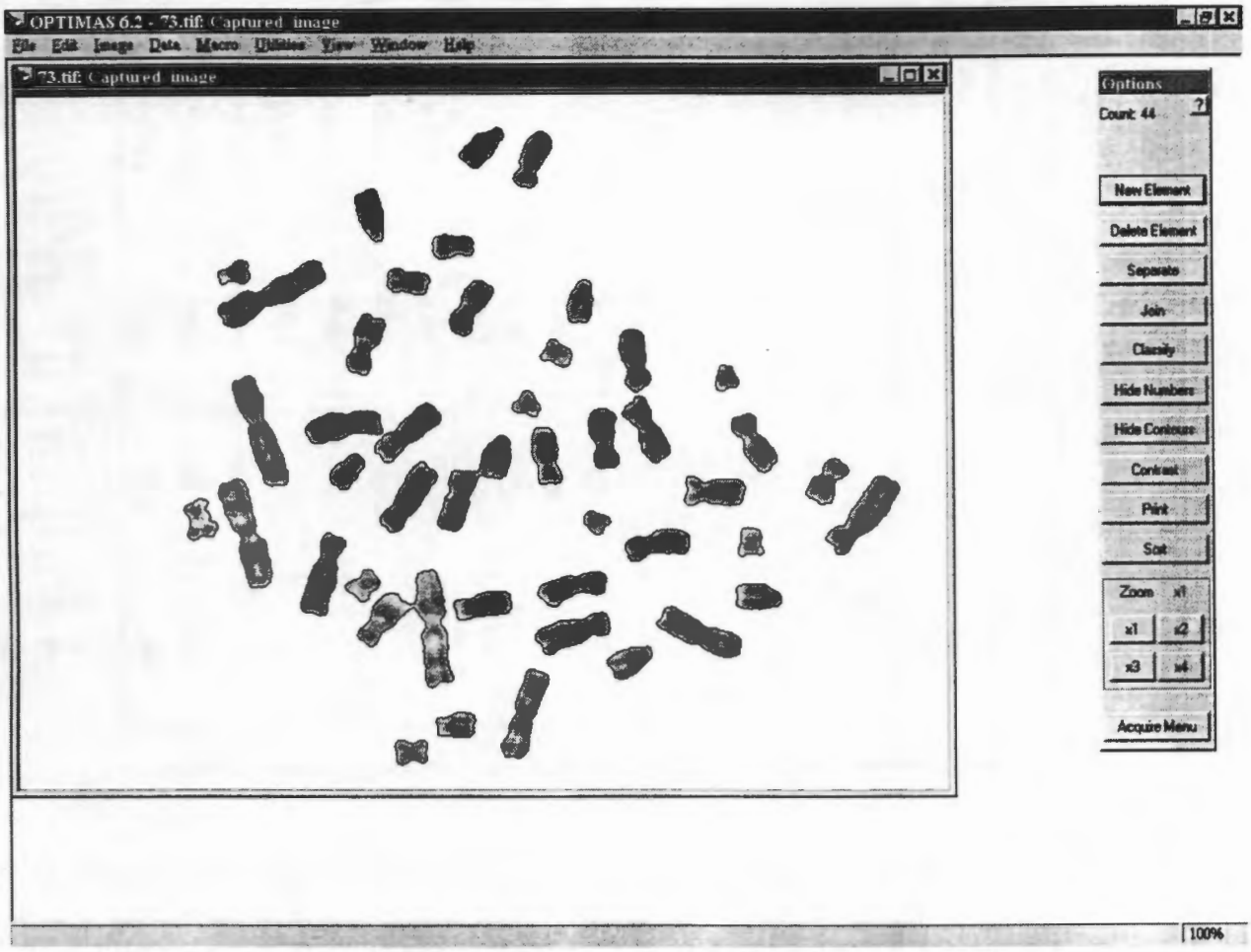


Figure 1.8: An image of a metaphase after it has been thresholded and the chromosomes automatically located. There are two pairs of chromosomes which have not been separated by thresholding, and these can be manually separated using the controls on the right.

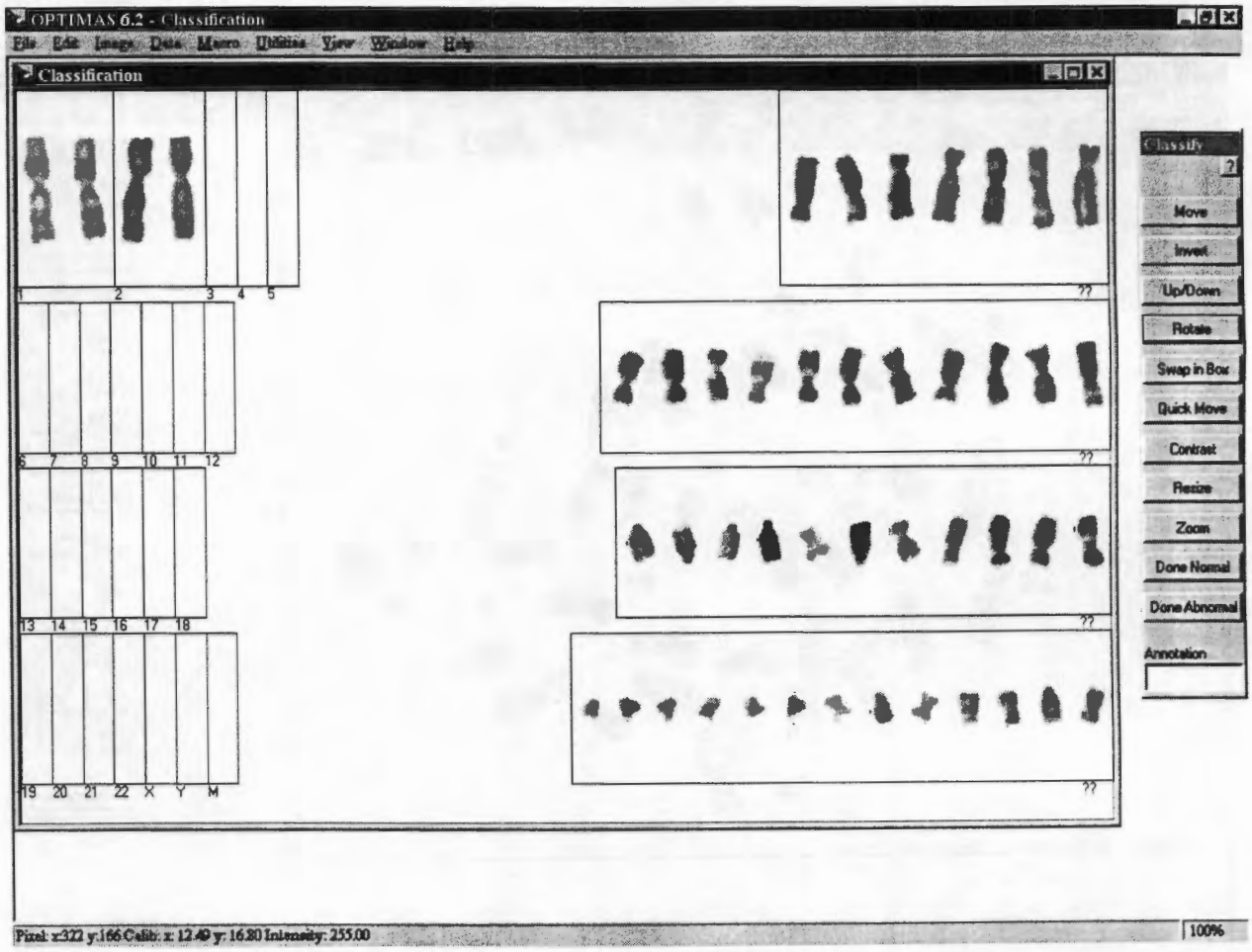


Figure 1.9: The Karyopt classification screen. The chromosomes are automatically sorted by length and appear in the boxes on the right of the screen. The user uses the controls on the right to move the chromosomes into the correct numbered boxes on the left, and to correct orientation and rotation. In this image, four chromosomes have already been moved into numbered boxes by the user.

Data Set	# Chroms	Remarks
Copenhagen (Cph)	8106	Peripheral blood cells, digitised by densitometry from photographic negatives
Edited Copenhagen	6985	Same as the Copenhagen set, but with overlapped and severely bent chromosomes excluded. It is described in detail by Lundsteen, Philip and Granum [49].
Edinburgh (Edi)	5548	Peripheral male blood cells, digitised by TV camera
Philadelphia (Phi)	5945	Chorionic villus cells digitised by CCD line scanner
600	6177	Long blood metaphases
Cpr	128990	Metaphase amnion cells
CDAR2	127925	Metaphase amnion cells

Table 1.3: Summary of standard data sets available for use in chromosome classification experiments. The second column shows the number of chromosomes in each data set.

The Cph, Edi, Phi, 600 and Cpr data sets are available on CD-ROM or via FTP from the United Kingdom Medical Research Council Human Genetics Unit in Edinburgh. Each data set is distributed in three formats:

1. Images – Each chromosome is saved as a grey-scale image.

2. Symbolic Profiles – The profile length, machine-found centromeric index, value showing whether the automatically determined chromosome orientation is correct, and profile extracted from each chromosome is stored in ASCII format. Profile extraction is described in Section 3.2. For the Cph data set, the position of the centromere marked by a cytogeneticist is also included.

3. Symbolic Feature Data – The values of thirty features extracted from each chromosome are stored in ASCII format. These features are described in detail in Chapter 3 (and are listed in Table 3.2).

These data sets are split into two sections, which allows uniformity amongst two-part cross-validation experiments performed by different groups.

Chapter 2

Statistical Pattern Recognition and Classification

This chapter gives an overview of several pattern recognition techniques that have been used for classifying chromosomes, and some of the techniques used in this dissertation. It covers parametric and non-parametric pattern recognition techniques and neural networks. A brief introduction to genetic algorithms is included. This is not strictly a pattern recognition technique, but has been used in order to optimise a set of classifications made by pattern recognition techniques. Readers can refer to Bishop [3], Duda and Hart [9], Fukunaga [13], Ripley [68], or Therrien [77] for a more thorough treatment of the subjects presented in this chapter.

2.1 Introduction

Pattern recognition is carried out almost effortlessly by the human brain. Humans can, for example, easily recognize faces and scenes, are able to learn to spot manufacturing faults on a production line, or to recognize unusually shaped objects on a microscope slide. Getting computers to perform the same tasks with the same efficiency has proved to be a difficult task. Most approaches to this problem have been within a statistical framework. This chapter concentrates on using statistical pattern recognition for the classification of objects.

Developing an automated classifier requires a training set, which is a set of example patterns preclassified into one of K predefined classes. These examples are used to develop a model of the underlying process which generated the example patterns. This model can then be used to classify patterns not in the training set.

The patterns are usually represented by a d-dimensional feature vector $\mathbf{X}$ which is made up of a set of numerical features extracted from the objects to be classified. These features should be chosen so as to provide good discrimination between patterns belonging

to different classes. The choice of features is problem dependent and they can be extracted in various ways depending on the form of the raw data. Sometimes the raw data itself is used as the set of features. In a practical application, the feature vectors are often corrupted by noise, which justifies the use of statistical methods of classification.

Ripley [67] gives a formal definition of classification as follows:

> We suppose there is a feature space $\mathcal{X}$ of potential observations, and to each we should assign a label, either that of one of K classes or 'doubt' $\mathcal{D}$ or 'outlier' $\mathcal{O}$. 'Doubt' is used when more than one class seems plausible and 'outlier' when no class is plausible. Let $\mathcal{Y} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K, \mathcal{D}, \mathcal{O}\}$ be the space of possible decisions; a *classifier* is then a map from $\mathcal{X}$ to $\mathcal{Y}$.
>
> This paradigm assumes cases ($\mathbf{X}$,Y) are drawn independently from a probability distribution; we call the random variable $\mathbf{X}$ the *pattern* and Y the true decision (which usually does not take the value $\mathcal{D}$). The *training set* $\mathcal{T}$ is a set of N classified cases. Future cases will present a pattern $\mathbf{X}$ for the classifier to compute a decision.

In the sections below, the doubt and outlier classes are sometimes combined into one class called the reject class $\mathcal{Z} = \mathcal{D} \cup \mathcal{O}$.

2.1.1 Model choice

Models are developed using the data in the training set, but should have the ability to *generalise* to unseen data. A pattern classification system which can classify all the examples in the training set correctly, but produces anomalous results when any new data is presented to it is of little use in a practical situation. Models should have enough flexibility to represent the data, but not *over-fit* to the data. In general, the number of parameters in the model should be less than the number of patterns in the training set.

Useful analogies can be drawn between modelling high-dimensional noisy feature vectors and fitting polynomials to one dimensional data corrupted by noise [3]. Consider fitting an Mth-order polynomial given by

$$y(x) = w_0 + w_1 x + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{2.1}$$

to a set of N points (x_i, y_i), where $i = 1, \ldots, N$, by finding a value for the w_j coefficients which minimises the sum of the squares of the differences between the fitted curve and each data point

$$E = \frac{1}{2} \sum_{i=1}^{N} \{y(x_i) - y_i\}^2 \tag{2.2}$$

As an example, a set of eleven points was generated by sampling the function

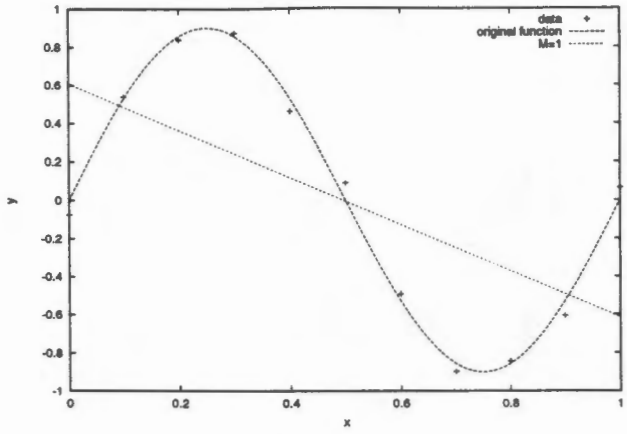


Figure 2.1: Eleven data points obtained by sampling the "original function" and adding noise. A fitted polynomial (equation 2.1) with $M = 1$ is shown.

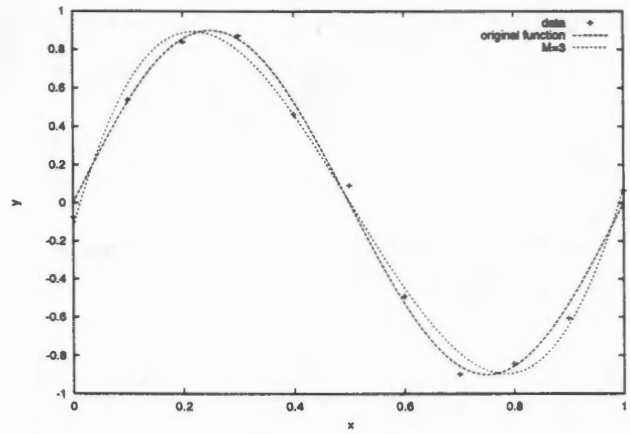


Figure 2.2: Eleven data points obtained by sampling the "original function" and adding noise. A fitted polynomial (equation 2.1) with $M = 3$ is shown.

$$f(x) = 0.9 \sin(2\pi x) \tag{2.3}$$

at equal intervals and adding a small random value to the y-coordinate of each point. Polynomial curves with $M = 1$, $M = 3$ and $M = 10$ were fitted to these points, and are shown in Figures 2.1, 2.2 and 2.3 respectively. The $M = 1$ polynomial is not flexible enough to model the data; the $M = 3$ polynomial provides good generalisation and is a convincing approximation to the underlying function; the $M = 10$ polynomial is overfitted to the data, as it fits the data points exactly, but does not generalise well between the data points.

2.2 Bayes decision theory

To classify a pattern vector $\mathbf{X}$ as belonging to a class $\mathcal{C}_k \in \mathcal{Y}$, it is necessary to know the posterior probability $P(\mathcal{C}_k|\mathbf{X})$, which is the probability of class $\mathcal{C}_k$ conditioned on (given)

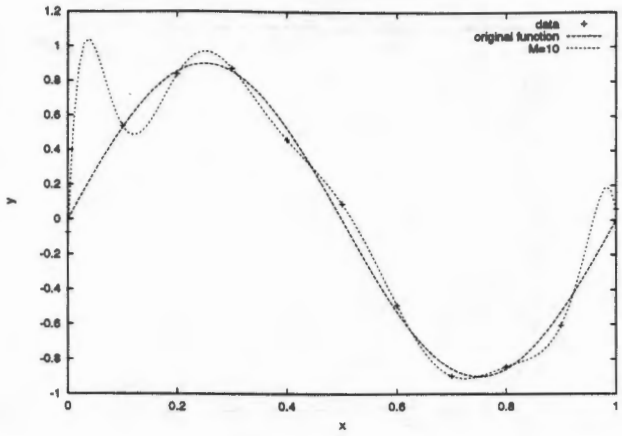


Figure 2.3: Eleven data points obtained by sampling the "original function" and adding noise. A fitted polynomial (equation 2.1) with $M = 10$ is shown.

$\mathbf{X}$, or the probability that an object with a feature vector $\mathbf{X}$ belongs to class $\mathcal{C}_k$. It is possible to write this in terms of parameters which are usually more easily estimated using Bayes' theorem

$$P(\mathcal{C}_k|\mathbf{X}) = \frac{p(\mathbf{X}|\mathcal{C}_k)\,P(\mathcal{C}_k)}{p(\mathbf{X})} \tag{2.4}$$

where $P(\mathcal{C}_k)$ is the prior probability of a pattern belonging to class $\mathcal{C}_k$, $p(\mathbf{X}|\mathcal{C}_k)$ is the class-conditional probability density[1] of $\mathbf{X}$ for class $\mathcal{C}_k$, and the unconditional density

$$p(\mathbf{X}) = \sum_{k=1}^{K} p(\mathbf{X}|\mathcal{C}_k)\,P(\mathcal{C}_k) \tag{2.5}$$

is a normalisation factor which ensures that

$$\sum_{k=1}^{K} P(\mathcal{C}_k|\mathbf{X}) = 1 \tag{2.6}$$

The prior probability is the probability that a pattern belongs to a certain class without considering the appearance of the pattern. For example, consider the process of classifying printed circuit boards at the end of a production line into one of two classes, "faulty" or "not-faulty". Depending on the efficiency and past history of the production line, one can assign a prior probability that a certain circuit board will be faulty before examining the board. The class-conditional probability density $p(\mathbf{X}|\mathcal{C}_k)$ is the probability density for vector $\mathbf{X}$ given that it is a member of class $\mathcal{C}_k$.

There are two separate stages in the classification process. The first is *inference*, in which data (the training set $\mathcal{T}$) are used to determine models for $p(\mathbf{X}|\mathcal{C}_k)$ and values for $P(\mathcal{C}_k)$, or sometimes to model $P(\mathcal{C}_k|\mathbf{X})$ directly. In the *decision making* process, these models are used to assign a new pattern to one of the classes.

[1] An upper-case P represents a ***probability*** and a lower-case p represents a ***probability density***. A probability density specifies that the probability that a vector variable $\mathbf{x}$ lies in the region $\mathcal{R}$ of $\mathbf{x}$-space is $P(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x})\,d\mathbf{x}$. The integral of $p(\mathbf{x})$ over the whole of $\mathbf{x}$-space is unity.

2.2.1 Inference

The values for the prior probability $P(\mathcal{C}_k)$ for each class are determined from the training set or from other prior knowledge. If it is known that objects from each class have an equal likelihood of appearing, then $P(\mathcal{C}_k)$ can be set equal to $\frac{1}{K}$. If the training set is completely representative of the number of objects in each class that will be encountered by the classifier when classifying new data, then the prior probabilities can be set by counting the number of objects belonging to each class in the training set. The prior probabilities can be used to correct for training sets which are not representative of the true ratios of the number of objects in a class. For example, if a classifier is trained to detect a certain fault on a manufactured object at the end of a production line, it is pertinent to have a large number of examples of faulty objects in the training set, even though a faulty object might occur only once in every thousand objects manufactured.

The conditional probability density $p(\mathbf{X}|\mathcal{C}_k)$ can be modelled using *parametric* or *non-parametric* estimation. For parametric methods, a specific functional form is assumed for the probability density, and the training set is used to optimize the parameters of the function. When using a non-parametric method, an attempt is made to model the probability density based solely on the training set with no assumptions made about underlying functional forms. Bishop [3] also mentions *semi-parametric* estimation which includes mixture models[2] and *neural networks*.

2.2.2 Decision making

When unseen feature vector $\mathbf{X}$ is presented to the trained classifier, the class to which $\mathbf{X}$ is assigned depends on the value of a set of *discriminant functions* $y_1(\mathbf{X}), \ldots, y_K(\mathbf{X})$, where $\mathbf{X}$ is assigned to class $\mathcal{C}_k$ if

$$y_k(\mathbf{X}) > y_j(\mathbf{X}) \qquad \text{for all } j \neq k \tag{2.7}$$

The most obvious definition for $y_k(\mathbf{X})$ is

$$y_k(\mathbf{X}) = P(\mathcal{C}_k|\mathbf{X}) = \frac{p(\mathbf{X}|\mathcal{C}_k)\,P(\mathcal{C}_k)}{p(\mathbf{X})} \tag{2.8}$$

as it is possible to show[3] that this minimises the average probability of error (the error rate). As the denominator of the expression on the right of equation 2.8 does not depend on $\mathbf{X}$, it is equivalent to set

$$y_k(\mathbf{X}) = p(\mathbf{X}|\mathcal{C}_k)\,P(\mathcal{C}_k)$$

As only the magnitude of $y_k(\mathbf{X})$ is important, it is also possible to use $g(y_k(\mathbf{X}))$, where g is any monotonic function.

[2] see Titterington et al. [79] and McLachlan and Basford [54]

[3] see Duda and Hart [9] pages 16–17

In some situations, the cost of assigning an object to a group varies depending on the group, for example, when inspecting safety fuses on explosive shells, it is far more serious to classify an armed fuse as unarmed than it is to classify an unarmed fuse as armed. To take this into account, a minimum risk classification[4] rather than a minimum error rate classification must be done. The classifier assigns a feature vector $\mathbf{X}$ to a class $\mathcal{C}_j$ if

$$\sum_{k=1}^{K} L_{kj} p\left(\mathbf{X}|\mathcal{C}_k\right) P\left(\mathcal{C}_k\right) < \sum_{k=1}^{K} L_{ki} p\left(\mathbf{X}|\mathcal{C}_k\right) P\left(\mathcal{C}_k\right) \qquad \text{for all } i \neq j \tag{2.9}$$

where L_{kj} are elements of the *loss matrix* specifying the penalty associated with assigning a pattern to class $\mathcal{C}_j$ when it belongs to class $\mathcal{C}_k$. If L_{kj} is set equal to $1 - \delta_{kj}$, where δ_{kj} is the Kronecker delta[5], then equation 2.9 reduces to equation 2.7 for minimum error rate classification.

It is often very difficult to decide when an unknown pattern should be assigned to the reject class $\mathcal{Z}$. An advantage of obtaining a good estimate of the posterior probability $P\left(\mathcal{C}_k|\mathbf{X}\right)$ is that a threshold θ can be set such that if $\max_k P\left(\mathcal{C}_k|\mathbf{X}\right) < \theta$ then $\mathbf{X}$ is assigned to the reject class. If a pattern is assigned to the reject class, it can then be further processed by another automated classifier or by a human.

2.3 Parametric methods

It is assumed that the functional form of $p\left(\mathbf{X}|\mathcal{C}_k\right)$ for each class $\mathcal{C}_k$ is known, and can be parametrised by a vector $\boldsymbol{\theta}_k$. The problem then reduces to determining the best values of $\boldsymbol{\theta}_k$ for each class using the samples in the training set. It is usually assumed that patterns in class $\mathcal{C}_j$ do not have any effect on $\boldsymbol{\theta}_k$ if $j \neq k$. To show the dependence of $p\left(\mathbf{X}|\mathcal{C}_k\right)$ on $\boldsymbol{\theta}_k$, $p\left(\mathbf{X}|\mathcal{C}_k\right)$ is written as $p\left(\mathbf{X}|\mathcal{C}_k, \boldsymbol{\theta}_k\right)$. Values of $\boldsymbol{\theta}_k$ can be determined by using a maximum likelihood method or Bayesian inference method. Let $\mathcal{X}^k = \left\{\mathbf{X}_1^k, \ldots, \mathbf{X}_m^k\right\}$ refer to the m samples in the training set of the class $\mathcal{C}_k$ under consideration. The most commonly used and simplest choice for a functional form of $p\left(\mathbf{X}|\mathcal{C}_k\right)$ is a multivariate Gaussian distribution

$$N\left(\boldsymbol{\mu}_k, \Sigma_k\right) = \frac{1}{(2\pi)^{d/2} \left|\Sigma_k\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{X} - \boldsymbol{\mu}_k\right)^{\mathrm{T}} \Sigma_k^{-1} \left(\mathbf{X} - \boldsymbol{\mu}_k\right)\right\} \tag{2.10}$$

with d-dimensional mean vector $\boldsymbol{\mu}_k$ and $d \times d$ covariance matrix Σ_k. The quantity

$$\Delta^2 = \left(\mathbf{X} - \boldsymbol{\mu}_k\right)^{\mathrm{T}} \Sigma_k^{-1} \left(\mathbf{X} - \boldsymbol{\mu}_k\right) \tag{2.11}$$

which appears in the exponent of equation 2.10 is called the *Mahalanobis distance* from $\mathbf{X}$ to $\boldsymbol{\mu}_k$, and is the Euclidean distance from $\mathbf{X}$ to $\boldsymbol{\mu}_k$ weighted by the covariance of the particular class. It is possible to use only the Mahalanobis distance when classifying an

[4] See Bishop [3] page 27

[5] $\delta_{jk} = 1$ if $j = k$ and 0 if $j \neq k$.

unknown pattern $\mathbf{X}$ by classifying it as belonging to the class having the smallest Mahalanobis distance between $\mathbf{X}$ and the class mean. This classifier is known as the *minimum Mahalanobis distance* classifier.

2.3.1 Maximum likelihood

As each sample is drawn independently from the distribution, the joint probability density of all the samples in class C_k is given by

$$p\left(\mathcal{X}^k \mid \boldsymbol{\theta}_k\right) = \prod_{l=1}^{m} p\left(\mathbf{X}_l^k \mid \boldsymbol{\theta}_k\right) \tag{2.12}$$

This is labelled $\mathcal{L}\left(\boldsymbol{\theta}_k\right)$, the *likelihood* of $\boldsymbol{\theta}_k$ with respect to the samples $\mathcal{X}^k$. The best value for $\boldsymbol{\theta}_k$ written as $\hat{\boldsymbol{\theta}}_k$ maximises equation 2.12. It is often easier to work with the logarithm of equation 2.12, called the log–likelihood

$$\ln \mathcal{L}\left(\boldsymbol{\theta}_k\right) = \ln p\left(\mathcal{X}^k \mid \boldsymbol{\theta}_k\right) = \sum_{l=1}^{m} \ln p\left(\mathbf{X}_l^k \mid \boldsymbol{\theta}_k\right) \tag{2.13}$$

For the general case, equation 2.12 or 2.13 must be maximised using an iterative numerical procedure, although it can be solved analytically for the multivariate gaussian [2] to give

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{m} \sum_{l=1}^{m} \mathbf{X}_l^k \tag{2.14}$$

and

$$\hat{\Sigma}_k = \frac{1}{m} \sum_{l=1}^{m} \left(\mathbf{X}_l^k - \hat{\boldsymbol{\mu}}_k\right)\left(\mathbf{X}_l^k - \hat{\boldsymbol{\mu}}_k\right)^{\mathrm{T}} \tag{2.15}$$

2.3.2 Bayesian inference

The goal of the maximum likelihood method is to find a single most likely parameter vector $\boldsymbol{\theta}_k$, while the goal of Bayesian methods is to find a probability density function for values of $\boldsymbol{\theta}_k$. In general, as more observations are added, the probability density of $\boldsymbol{\theta}_k$ becomes more sharply peaked at the position of the maximum likelihood value of $\boldsymbol{\theta}_k$. Thus, for large values of N the two methods give very similar results[6].

2.4 Non-parametric methods

Let $\hat{\mathbf{x}}$ be a point in a multi-dimensional space at which the probability density $p\left(\hat{\mathbf{x}}\right)$ must be estimated, and let $\mathcal{R}$ be a region surrounding $\hat{\mathbf{x}}$. The probability that a vector $\tilde{\mathbf{x}}$ drawn

[6]For a detailed account on using Bayesian inference to estimate parameters, see Duda and Hart [9], pages 49–59.

from the probability density $p(\mathbf{x})$ is in $\mathcal{R}$ is by definition

$$P = \int_{\mathcal{R}} p(\mathbf{x})\, d\mathbf{x} \tag{2.16}$$

Using the mean-value theorem, this can be written as

$$P = p(\mathbf{x}')\, V \tag{2.17}$$

where $\mathbf{x}'$ is a suitable point in $\mathcal{R}$ and $V = \int_{\mathcal{R}} d\mathbf{x}$ is the volume of $\mathcal{R}$. If $p(\mathbf{x})$ is continuous and $\mathcal{R}$ is small enough, then $p(\mathbf{x}')$ can be replaced by $p(\hat{\mathbf{x}})$, so one can write

$$p(\hat{\mathbf{x}}) = \frac{P}{V} \tag{2.18}$$

If N points are drawn independently from $p(\mathbf{x})$, then the probability that k of them will fall into $\mathcal{R}$ is given by the binomial distribution

$$\Pr(k) = \frac{N!}{k!\,(N-k)!} P^k (1-P)^{N-k} = \binom{N}{k} P^k (1-P)^{N-k} \tag{2.19}$$

The mean (expectation value) and variance of a binomial distribution are

$$\mathcal{E}[k] = NP \tag{2.20}$$

and

$$\mathrm{Var}(k) = \mathcal{E}\left[k^2\right] - (\mathcal{E}[k])^2 = NP(1-P) \tag{2.21}$$

To find $\hat{P}$, the maximum likelihood estimate of P, equation 2.19 is differentiated with respect to P and set equal to zero

$$\begin{aligned}
\frac{d\Pr(k)}{dP} &= \binom{N}{k} \left[k P^{k-1} (1-P)^{N-k} - (N-k) P^k (1-P)^{N-k-1}\right] \\
&= \binom{N}{k} P^{k-1} (1-P)^{N-k-1} \left[K(1-P) - (N-k)P\right] \\
&= 0
\end{aligned}$$

Solving for $\hat{P}$, one obtains

$$\begin{aligned}
k\left(1-\hat{P}\right) &= (N-k)\hat{P} \\
\Rightarrow \quad \hat{P} &= \frac{k}{N}
\end{aligned} \tag{2.22}$$

This estimate is unbiased as $\mathcal{E}\left[\hat{P}\right] = \frac{\mathcal{E}[k]}{N} = P$ (from equation 2.20).

The variance of $\hat{P}$

$$\begin{aligned}
\mathrm{Var}\left(\hat{P}\right) &= \mathcal{E}\left[\hat{P}^2\right] - \left(\mathcal{E}\left[\hat{P}\right]\right)^2 \\
&= \frac{\mathcal{E}\left[k^2\right] - (\mathcal{E}[k])^2}{N^2} \\
&= \frac{P(1-P)}{N}
\end{aligned}$$

approaches 0 as $N \rightarrow \infty$. Substituting equation 2.22 into equation 2.18 produces the estimate of the probability density at point $\hat{\mathbf{x}}$

$$p(\hat{\mathbf{x}}) = \frac{k}{NV} \tag{2.23}$$

There are now two possible ways of proceeding. One can keep V constant and count the number of pattern vectors in V, or one can change V so that it encloses k pattern vectors. The first approach leads to the Parzen window or kernel based methods and the second approach leads to k-nearest neighbour methods.

2.4.1 Kernel based methods

The simplest form of this method involves placing a d-dimensional hypercube (where d is the number of features in a pattern vector) with sides of length h centered at the point $\hat{\mathbf{x}}$ at which the density estimate is to be made. The volume of the cube is

$$V = h^d \tag{2.24}$$

One can define a *kernel function* $\mathcal{H}(\mathbf{u})$ as

$$\mathcal{H}(\mathbf{u}) = \begin{cases} 1 & \text{if } |u_j| < \frac{1}{2} \quad j = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

so that $\mathcal{H}\left(\frac{\mathbf{X}_l - \hat{\mathbf{x}}}{h}\right)$ is 1 if a pattern $\mathbf{X}_l$ from the training set lies within the hypercube centered at $\hat{\mathbf{x}}$. The number of patterns within the hypercube can be calculated as

$$k = \sum_{l=1}^{N} \mathcal{H}\left(\frac{\mathbf{X}_l - \hat{\mathbf{x}}}{h}\right) \tag{2.26}$$

The density estimate at point $\hat{\mathbf{x}}$ can be obtained by combining equations 2.23, 2.24 and 2.26 to get

$$p(\hat{\mathbf{x}}) = \frac{1}{N} \sum_{l=1}^{N} \frac{1}{h^d} \mathcal{H}\left(\frac{\mathbf{X}_l - \hat{\mathbf{x}}}{h}\right) \tag{2.27}$$

It is equivalent to visualise this as placing a hypercube of side length h at each point $\mathbf{X}_l$ in the training set and counting the number of cubes that contain $\hat{\mathbf{x}}$. A smoother kernel function such as a Gaussian distribution is normally used to provide a smoother probability density estimate. Equation 2.27 then becomes

$$p(\hat{\mathbf{x}}) = \frac{1}{N} \sum_{l=1}^{N} \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left\{-\frac{\|\mathbf{X}_l - \hat{\mathbf{x}}\|}{2\sigma^2}\right\} \tag{2.28}$$

where σ is the width of the Gaussian kernel. It should be noted that these kernel based methods make the assumption that all the components of the pattern vectors have an equal amount of spread h or an equal standard deviation σ. It is also assumed that the components of the pattern vectors are independent and that the covariance matrix is therefore diagonal.

2.4.2 k-nearest neighbour methods

If k is kept fixed and V is taken to be the volume of a hypersphere centred at $\hat{\mathbf{x}}$ and just large enough to contain precisely k training set points, then $p(\hat{\mathbf{x}})$ can be estimated directly using equation 2.23.

The k-nearest neighbour method can also be used to construct a classifier directly. N hyperspheres, each encompassing k training set points from each class are placed at point $\mathbf{X}$. $\mathbf{X}$ is then classified as belonging to the class having the smallest volume hypersphere.

The more commonly used approach is to use a hypersphere centered at $\mathbf{X}$ with volume V containing exactly k training set points of any class. The number of points k_l belonging to each class C_l within the hypersphere is then counted. The class conditional density $p(\mathbf{X}|C_l)$ is estimated as

$$p(\mathbf{X}|C_l) = \frac{k_l}{N_l V} \tag{2.29}$$

where N_l is the number of points in class C_l, the prior probability $P(C_l)$ is estimated as

$$p(C_l) = \frac{N_l}{N} \tag{2.30}$$

and the unconditional density is obtained from equation 2.23. Substituting equations 2.23, 2.29 and 2.30 into Bayes' theorem (equation 2.4) results in

$$p(C_l|\mathbf{X}) = \frac{k_l}{k} \tag{2.31}$$

Thus, to minimise the probability of misclassifying an unknown pattern $\mathbf{X}$, it should be assigned to the class for which k_l is largest. This is known as the *k-nearest neighbour classification rule.* The special case $k = 1$ is known as the *nearest neighbour rule.*

2.4.3 Advantages and disadvantages of these methods

The chief disadvantage of both methods is that all the training set examples have to be stored, and the distances between a point to be classified and all the stored points must be computed whenever a pattern vector is classified. There are a number of algorithms which allow the number of stored data points to be reduced[7].

Another disadvantage of the kernel-based approach is the difficulty in choosing the kernel size parameter h. Choosing an overly large size parameter can smooth the density estimate in some regions of feature space. A small value of h can lead to noisy probability density estimates in regions of feature space with a low density of training set points. The k-nearest neighbour methods do not suffer from these drawbacks to the same extent, although small values of k will likewise lead to noisy classifications, and the probability density is in effect assumed constant over the minimum sphere containing the k training points, which may be large in sparse regions.

[7]See, for example, Fukunaga [13] pages 359–362

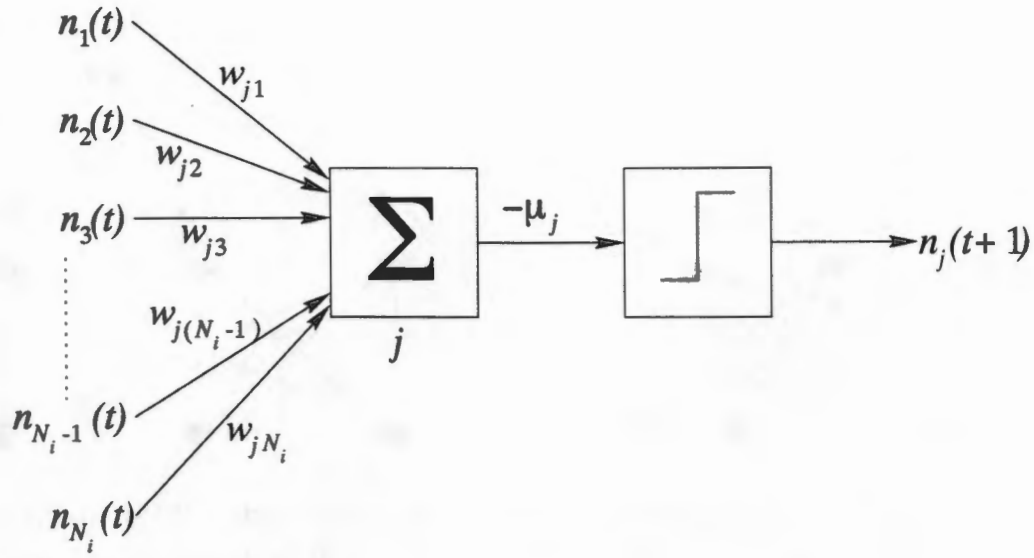


Figure 2.4: A McCulloch-Pitts neuron

2.5 Artificial neural networks

2.5.1 A brief history

Artificial neural networks (ANNs) were originally developed as simple models of the human brain, which consists of the order of 10^{11} interconnected neurons. There have been three main surges of interest in ANNs.

In 1943, McCulloch and Pitts [53] proposed the simple threshold unit shown in Figure 2.4 as a model of a neuron (this later became known as the McCulloch-Pitts neuron). A neuron accepts a number of binary inputs, and outputs a 1 or 0 according to the following formula

$$n_j(t+1) = H\left(\sum_{i=1}^{N_i} w_{ji} n_i(t) - \mu_j\right) \tag{2.32}$$

where $n_i(t)$ is the value of input i at time t, $n_j(t+1)$ is the output of neuron j at the next time iteration, w_{ji} is the weight on connection i to neuron j, N_i is the number of inputs to neuron j, μ_j is the threshold or bias associated with neuron j and H is the Heaviside function

$$H(\theta) = \begin{cases} 1 \text{ if } \theta \geq 0 \\ 0 \text{ if } \theta < 0 \end{cases} \tag{2.33}$$

McCulloch and Pitts showed that any finite logical expression can be realised by a network of McCulloch-Pitts neurons [1].

Around 1960, Rosenblatt and his group focussed on neural network models called *perceptrons* [70]. These were layers of McCulloch-Pitts neurons with the outputs of one layer feeding into the inputs of the next layer. A single layer perceptron (simple perceptron) and a two layer perceptron are shown in Figure 2.5. Traditionally, these would be referred to as

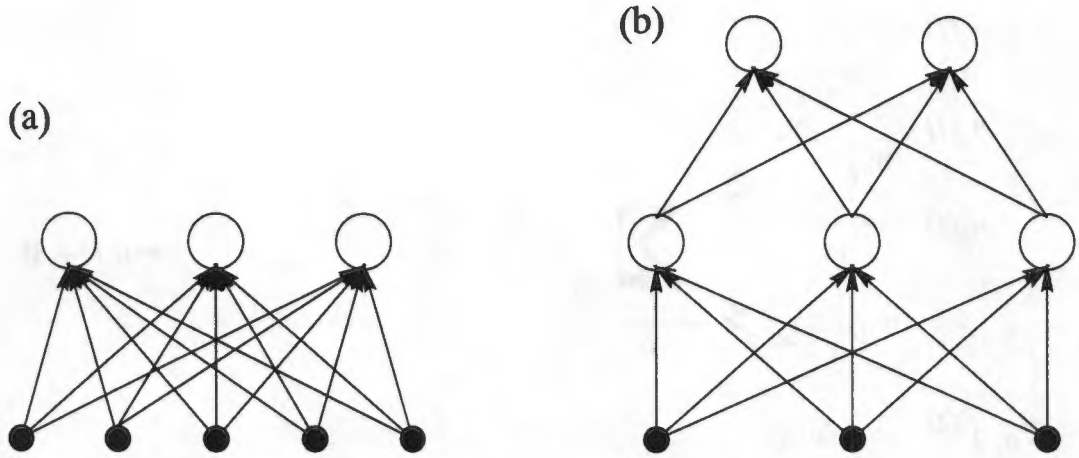


Figure 2.5: Perceptrons. (a) A one-layer perceptron (simple perceptron). (b) A two-layer perceptron. The inputs are shown as small filled circles to differentiate them from the other units, as no processing is performed by the inputs. The inputs are therefore not counted as a layer.

two and three layer perceptrons, but the modern convention is not to count the inputs as a layer as they simply pass data straight onto the next layer without processing it in any way. Input data is presented to the network inputs, and the outputs are read from the outputs of the last layer of neurons. Similar networks called *adalines* were invented by Widrow and Hoff in the same period [85]. An iterative algorithm called the *perceptron learning rule* was developed to train simple perceptrons (by adjusting the weights) using examples of input and associated output patterns. It can be proved that if there is a solution to the problem, then the perceptron learning rule will converge to the solution in a finite number of steps [4] [70]. A similar learning rule for perceptrons with more than one layer was not developed. In 1969, Minsky and Papert [56] pointed out a number of limitations of simple perceptrons. Their main criticism was that simple perceptrons could only learn linearly separable problems. For example, it is impossible for a simple perceptron having two inputs and one output to learn to model a boolean XOR operation. After the publication of this work, research interest in neural networks dampened for about 20 years.

In 1986, Rumelhart, Hinton and Williams [71] [72] popularised the back propagation algorithm, which allows training of networks with an arbitrary number of layers (this algorithm is described and derived in Section 2.5.3). This revived interest in neural networks.

2.5.2 The multilayer perceptron (MLP)

In order for the back propagation algorithm to be used, the McCulloch-Pitts neurons are replaced by generalised units[8]. The differences are

[8]The neurons are referred to as units to emphasize that the MLP is not considered to be a good model of a human brain.

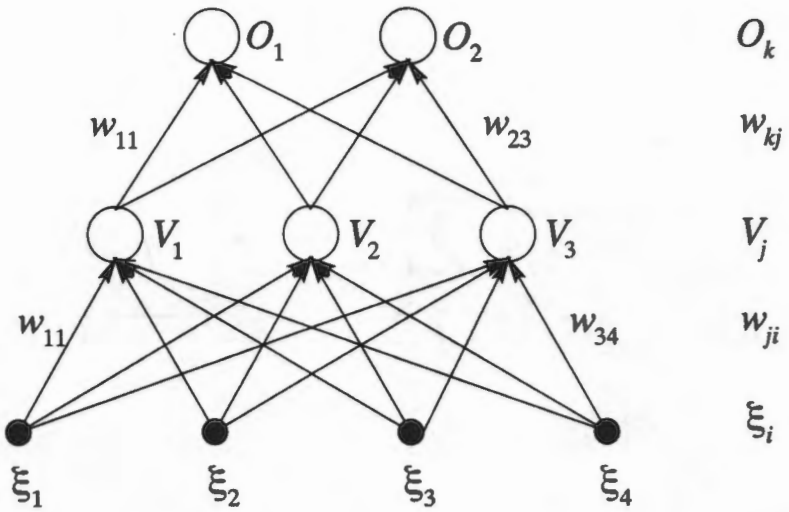


Figure 2.6: A two–layer MLP showing the notation used for weights and unit activations.

- The inputs and outputs are continuous valued variables.
- The Heaviside function (equation 2.33) is replaced by a non-linear function $g(x)$ called the *activation function*, *gain function*, *transfer function* or *squashing function*. This change is needed as the activation functions must be differentiable to allow the back propagation algorithm to be used.
- The times t and $t+1$ do not appear in the updating equation. This allows the units to be updated in any fixed or random order.

Figure 2.6 shows the nomenclature used to refer to the units and weights of an MLP. ξ_i refers to the value of the ith input unit out of a total of N_i, V_j refers to the activation of the jth hidden layer unit of N_j, and O_k refers to the activation of the kth output unit of N_k. The units in each layer are numbered starting at 1. The weight connecting unit i to unit j is referred to as w_{ji}. The activation functions of the hidden layer units are $g_h(x)$ and the activation functions of the output units are $g_o(x)$.

A hidden layer unit is shown in Figure 2.7. The updated version of equation 2.32 for this unit is

$$V_j = g_h\left(\sum_{i=1}^{N_i} w_{ji}\xi_i - \mu_j\right) \tag{2.34}$$

and the updated version for an output unit is

$$O_k = g_o\left(\sum_{j=1}^{N_j} w_{kj}V_j - \mu_k\right) \tag{2.35}$$

The above two equations can be simplified by replacing the bias terms μ_j and μ_k by weights w_{i0} connected to unit 0 which has an activation of -1. Equation 2.34 can then be

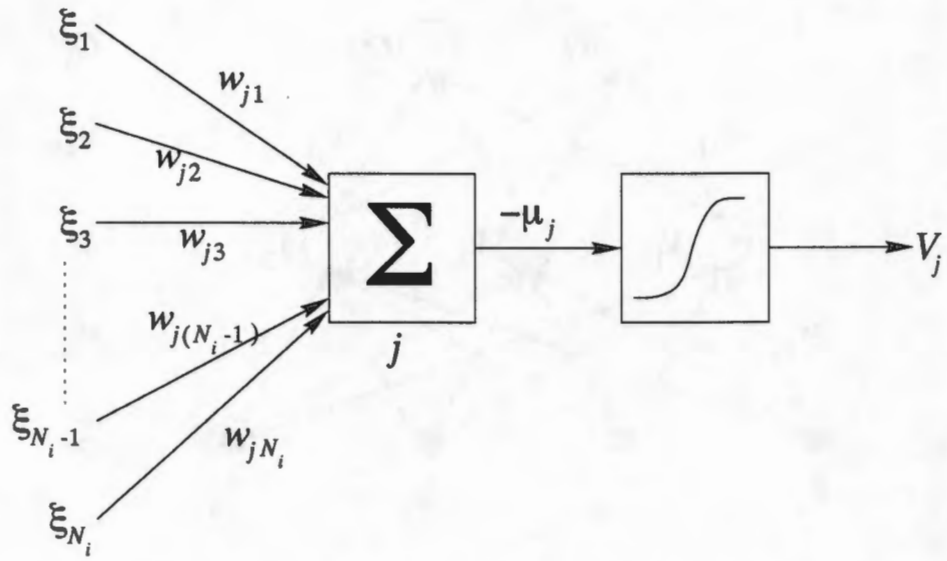


Figure 2.7: A hidden layer unit in an MLP

written as

$$V_j = g_h\left(\sum_{i=0}^{N_i} w_{ji}\xi_i\right) = g_h\left(h_j\right) \tag{2.36}$$

and equation 2.35 can be written as

$$O_k = g_o\left(\sum_{j=0}^{N_j} w_{kj}V_j\right) = g_o\left(h_k\right) \tag{2.37}$$

where h_j and h_k are a convenient notation for the weighted sums of the inputs to units j and k respectively. The advantage of this simplification is that the bias can be treated as an ordinary weight during training. For the network in Figure 2.6, the output activation of a unit in the output layer can be obtained by combining equations 2.36 and 2.37 to obtain

$$O_k = g_o\left(\sum_{j=0}^{N_j} w_{kj}g_h\left(\sum_{i=0}^{N_i} w_{ji}\xi_i\right)\right) \tag{2.38}$$

Activation functions

At least one layer of neurons in the MLP must have a non–linear activation function, else the network is equivalent to a one layer network which performs linear regression. The most common activation functions for the hidden and output layers are the sigmoid function

$$g\left(x\right) = \frac{1}{1+\exp\left(x\right)} \tag{2.39}$$

and the tanh function

$$g\left(x\right) = \tanh\left(x\right) \tag{2.40}$$

which are sketched in Figure 2.8. Linear activation functions are often used for the output units as they don't restrict the range of values that can be produced by these units.

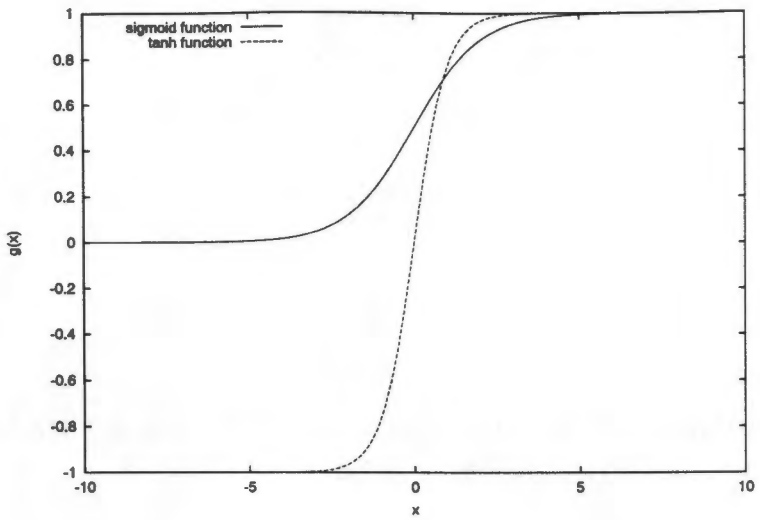


Figure 2.8: The sigmoid function (equation 2.39) and the tanh function (equation 2.40).

2.5.3 The back propagation algorithm

The back propagation algorithm is an iterative algorithm used to train MLPs. The derivation below is based on the derivation in [27]. The training set consists of N pairs of N_i-dimensional input vectors $\boldsymbol{\xi}_l$ and corresponding N_k-dimensional target output vectors $\boldsymbol{\zeta}_l$, where $1 \leq l \leq N$, and $\boldsymbol{\zeta}_l$ represents what the network output should be when $\boldsymbol{\xi}_l$ is presented to the inputs. Component j of input vector l is denoted as ξ_j^l. An error function E^l is defined which quantifies the difference between the output vector $\mathbf{O}_l$ produced by the network when vector $\boldsymbol{\xi}_l$ is presented to the inputs, and the target output vector $\boldsymbol{\zeta}_l$. Possible forms of this error function are discussed below, but for this derivation it is taken to be a general function of $\mathbf{O}_l$ and $\boldsymbol{\zeta}_l$.

One iteration (epoch) of the back propagation algorithm involves presenting all the patterns in the training set to the network and calculating the error and error gradient. The weights w_{ji} and w_{kj} are then adjusted to decrease the error.

The total error E after all the patterns in the training set have been presented to the network is simply

$$E = \sum_l E^l \tag{2.41}$$

The back propagation algorithm aims to minimise E. The partial derivative of the error E with respect to a weight w_{kj} leading to the output layer is calculated from equations 2.37 and 2.41.

$$\begin{aligned} \frac{\partial E}{\partial w_{kj}} &= \sum_l \frac{\partial E^l}{\partial O_k^l} \frac{\partial O_k^l}{\partial w_{kj}} \\ &= \sum_l \frac{\partial E^l}{\partial O_k^l} \frac{\partial O_k^l}{\partial h_k^l} \frac{\partial h_k^l}{\partial w_{kj}} \end{aligned}$$

$$
\begin{aligned}
&= \sum_l \frac{\partial E^l}{\partial O_k^l} \frac{\partial O_k^l}{\partial h_k^l} V_j^l \\
&= \sum_l \frac{\partial E^l}{\partial O_k^l} g_o' \left(h_k^l\right) V_j^l \\
&= \sum_l \delta_k^l V_j^l
\end{aligned} \tag{2.42}
$$

where

$$
\delta_k^l = \frac{\partial E^l}{\partial O_k^l} g_o' \left(h_k^l\right) \tag{2.43}
$$

The partial derivative of the error E with respect to a weight w_{ji} leading to the hidden layer is calculated from equations 2.38 and 2.41.

$$
\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \sum_l \sum_k \frac{\partial E^l}{\partial O_k^l} \frac{\partial O_k^l}{\partial w_{ji}} \\
&= \sum_l \sum_k \frac{\partial E^l}{\partial O_k^l} \frac{\partial O_k^l}{\partial h_k^l} \frac{\partial h_k^l}{\partial V_j^l} \frac{\partial V_j^l}{\partial w_{ji}} \\
&= \sum_l \sum_k \frac{\partial E^l}{\partial O_k^l} g_o' \left(h_k^l\right) w_{kj} g_h' \left(h_j^l\right) \xi_i^l \\
&= \sum_l \sum_k \delta_k^l w_{kj} g_h' \left(h_j^l\right) \xi_i^l \\
&= \sum_l \delta_j^l \xi_i^l
\end{aligned} \tag{2.44}
$$

where

$$
\delta_j^l = g_h' \left(h_j^l\right) \sum_k \delta_k^l w_{kj} \tag{2.45}
$$

It is simple to implement a gradient descent algorithm which adjusts the weights to decrease E. Once the partial derivative of E with respect to each weight has been calculated, each weight leading to the output units is updated using $w_{kj}^{\text{new}} = w_{kj}^{\text{old}} + \Delta w_{kj}$, where

$$
\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \tag{2.46}
$$

and each weight leading to a hidden unit is updated using $w_{ji}^{\text{new}} = w_{ji}^{\text{old}} + \Delta w_{ji}$, where

$$
\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \tag{2.47}
$$

and η is a constant known as the learning rate. At each iteration, this algorithm takes a step in the direction of the steepest descent on the error surface. The value of η is determined by experience and experiment, and is usually set to a value between 0 and 1. Gradient descent is not the most efficient non-linear optimisation algorithm, and the choice of η is generally very difficult (the optimum choice of η tends to vary during the training procedure). Another problem with this algorithm is that the direction towards the minimum

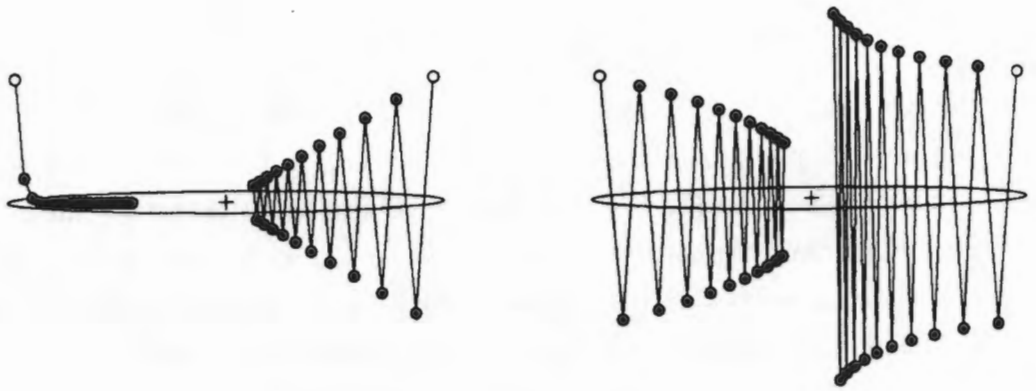


Figure 2.9: Gradient descent on a simple quadratic surface $E = x^2 + 20y^2$ (the left and right diagrams are copies of the same surface). The minimum is at + and the ellipse shows a constant error contour. Four trajectories of 20 iterations starting at the open circle are shown. The only significant difference between the trajectories is the value of η, which is 0.02, 0.0476, 0.049, and 0.0505 from left to right (from [27]).

is not always in the direction of the steepest descent, which can lead to oscillations. This is illustrated in Figure 2.9, which shows the first 20 iterations taken by the gradient descent algorithm when finding the minimum of a simple quadratic error surface $E = x^2 + 20y^2$ for four values of η. The error surfaces encountered when training neural networks tend to be very complicated with many local minima and other undesirable qualities. It is generally very difficult to decide whether an optimisation algorithm has reached the global minimum or become trapped in a local minimum.

It is possible to remove the sums over all the patterns in the training set in equations 2.42 and 2.44 and adjust the weights after each pattern has been presented. There are a number of more efficient algorithms for training MLPs which are described in Section 2.5.5.

Error functions

The most common error function used in training an MLP is the sum of squares

$$E^l = \frac{1}{2} \sum_{k=1}^{N_k} \left(\zeta_k^l - O_k^l\right)^2 \tag{2.48}$$

If this error function is used, then $\frac{\partial E^l}{\partial O_k^l} = -\left(\zeta_k^l - O_k^l\right)$ and $\delta_k^l = -\left(\zeta_k^l - O_k^l\right) g_o'\left(h_k^l\right)$. A large number of possible error functions are described by Masters [51][9].

[9]See pages 38–55

2.5.4 Using neural networks for pattern recognition

When using neural networks for pattern recognition, the usual approach is to have one input unit for each feature in the input vector $\mathbf{X}$. For a problem with only two classes, it is possible to use one output unit with a target activation $\zeta = 1$ for input vectors belonging to class $\mathcal{C}_1$ and $\zeta = 0$ for class $\mathcal{C}_2$. For a problem involving more than two classes, a 1-of-K encoding scheme for the output units is usually adopted, where the target value of each output unit k when an input vector $\mathbf{X}_l$ belonging to class $\mathcal{C}_j$ is presented to the network is $\zeta_k^l = \delta_{jk}$ (this means that each class is represented by one output unit).

Bishop [3] demonstrates[10] that if one uses the *cross-entropy* error function

$$E = -\sum_{l=1}^{N}\sum_{k=1}^{N_k} \zeta_k^l \ln\left(\frac{O_k^l}{\zeta_k^l}\right) \tag{2.49}$$

and softmax output activation functions

$$g_o\left(O_k\right) = \frac{\exp\left(O_k\right)}{\sum_{k'} \exp\left(O_{k'}\right)} \tag{2.50}$$

where the sum in the denominator is over all output units, then the outputs represent the posterior probabilities that an input vector $\mathbf{X}$ belongs to each class $\mathcal{C}_k$, or

$$O_k = p\left(\mathcal{C}_k|\mathbf{X}\right) \tag{2.51}$$

with the additional useful property that

$$\sum_{k=1}^{N_k} O_k = 1 \tag{2.52}$$

2.5.5 Training algorithms

There have been a large number of heuristic adjustments to equations 2.46 and 2.47 to improve the speed and efficiency of learning, and the discovery that a number of traditional numerical optimisation algorithms can be applied to neural network learning. Gibb [15] provides a comprehensive overview of neural network training methods. In this section, brief introductions are given to "Backpropagation with momentum", as it is the most ubiquitous training algorithm; and "Conjugate Gradient Methods" and "Simulated Annealing", as these are used to train the networks used for the experiments in Chapter 6. Genetic algorithms (Section 2.7) can also be used to train neural networks [50].

Backpropagation with momentum

A simple way of dealing with the oscillating trajectories sometimes exhibited by the gradient descent algorithm (shown in Figure 2.9) is the addition of a momentum term. This involves

[10] See pages 237–240

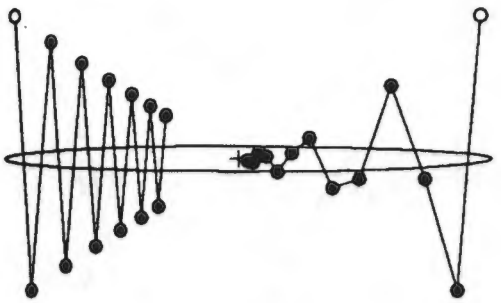


Figure 2.10: Gradient descent on the simple quadratic surface of Figure 2.9. Both trajectories are for 12 steps with $\eta = 0.0476$. On the left there is no momentum ($\alpha = 0$), while $\alpha = 0.5$ on the right (from [27]).

modifying equations 2.46 and 2.47 to have the form

$$\Delta w_{pq}(t+1) = -\eta \frac{\partial E}{\partial w_{pq}} + \alpha \Delta w_{pq}(t) \tag{2.53}$$

where α is the *momentum parameter*, which must have a value of between 0 and 1. The idea behind this extra term is to give each weight w_{pq} some "momentum" or "inertia" based on the direction moved in the previous iteration, so that large oscillations are damped out and the algorithm proceeds sedately down the hill in the "average" direction.

The effect of the momentum term can be demonstrated by first considering motion through a section of relatively low curvature. The gradient term $\frac{\partial E}{\partial w_{pq}}$ will remain almost constant, and after a number of iterations, equation 2.53 will become

$$\begin{aligned} \Delta w_{pq} &= -\eta \frac{\partial E}{\partial w_{pq}} \left(1 + \alpha + \alpha^2 + \ldots\right) \\ &= -\frac{\eta}{1-\alpha} \frac{\partial E}{\partial w_{pq}} \end{aligned} \tag{2.54}$$

increasing the effective learning rate from η to $\frac{\eta}{1-\alpha}$. On steep oscillatory sections, successive momentum terms will tend to cancel, making the effective learning rate close to η. A demonstration of the effectiveness of gradient descent with momentum is shown in Figure 2.10.

Conjugate gradient algorithm

The conjugate gradient algorithm has the advantage that no parameters have to be set by the user. This solves one of the difficulties involved in using the gradient descent with momentum algorithm, which is the choice of the step-size and momentum parameters η and α. The conjugate gradient algorithm uses a series of line minimisations to find the minimum of the error function. Each iteration τ consists of finding the value of λ which minimises the error function along a line

$$\mathbf{x} = \mathbf{x}_0^{(\tau)} + \lambda \mathbf{d}^{(\tau)} \tag{2.55}$$

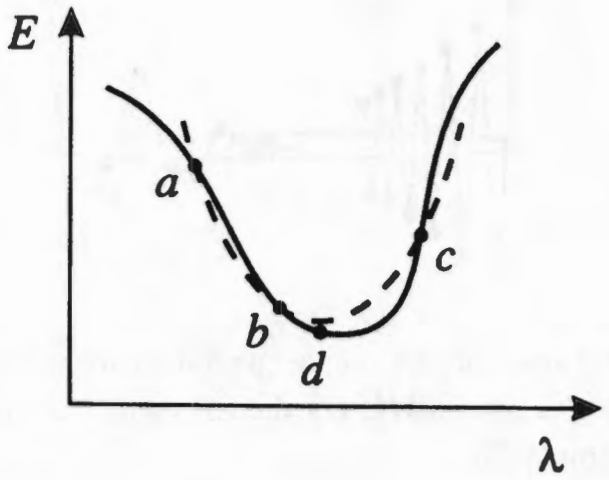


Figure 2.11: An illustration of the first step in a simple line minimisation process. The solid curve shows the error as a function of distance λ along the search direction. Three points $a < b < c$ are chosen and a parabola (dotted curve) is fitted to $E(a)$, $E(b)$ and $E(c)$. The minimum of the parabola at point d gives an approximation to the minimum of the error (from [3]).

starting at point $\mathbf{x}_0^{(\tau)}$ in a direction $\mathbf{d}^{(\tau)}$, and then choosing a direction $\mathbf{d}^{(\tau+1)}$ for a new line starting at the minimum found on the previous line.

The basis of the method used to find the λ which minimises equation 2.55 is to choose three points $a < b < c$ on the line such that $E(a) > E(b)$ and $E(c) > E(b)$, fit a parabola to these points, choose the minimum of the parabola as point d, fit a parabola to the three points which have the smallest function values, and repeat the process until the required termination criteria are reached. Figure 2.11 illustrates one step of this process. In practice, there are a number of refinements made to this method leading to the very robust Brent's method described in Press et. al. [66].

Unfortunately, the intuitively obvious choice for the new line direction $\mathbf{d}^{(\tau+1)}$ at each minimum point — the direction of steepest descent — proves to be a bad choice, as this direction will always be perpendicular to the direction of the line used in the previous step. If it were not perpendicular then there would still be a non-zero derivative in the direction $\mathbf{d}^{(\tau)}$. One can therefore write

$$\mathbf{d}^{(\tau)} \cdot \nabla E^{(\tau+1)} = 0 \tag{2.56}$$

where $\nabla E^{(\tau+1)}$ is the error function gradient at the minimum of the line with direction $\mathbf{d}^{(\tau)}$. Using the direction of steepest descent as a new direction would lead to oscillations in the trajectory followed by the algorithm.

A better method is to choose the new direction $\mathbf{d}^{(\tau+1)}$ as a compromise between the previous search direction and the gradient at the new point

$$\mathbf{d}^{(\tau+1)} = -\nabla E^{(\tau+1)} + \beta_\tau \mathbf{d}^{(\tau)} \tag{2.57}$$

The value of β_τ should be chosen so as not to change (to first order) the component of the gradient parallel to the direction $\mathbf{d}^{(\tau)}$, which was just made zero. This requires that

$$\mathbf{d}^{(\tau)} \cdot \nabla E\left(\mathbf{x}_0^{(\tau+1)} + \lambda \mathbf{d}^{(\tau+1)}\right) = 0 \tag{2.58}$$

Expanding the error function $E(\mathbf{x})$ about $\mathbf{x}_0$ as

$$E(\mathbf{x}) = E_0 + (\mathbf{x} - \mathbf{x}_0)\nabla E(\mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{H} \cdot (\mathbf{x} - \mathbf{x}_0) + \ldots \tag{2.59}$$

where $\mathbf{H}$ is the second derivative *Hessian Matrix*

$$\mathbf{H}_{ij} = \frac{\partial^2 E}{\partial x_i \partial x_j} \tag{2.60}$$

evaluated at $\mathbf{x}_0$, differentiating it to get

$$\nabla E(\mathbf{x}) = \nabla E(\mathbf{x}_0) + \mathbf{H} \cdot (\mathbf{x} - \mathbf{x}_0) + \ldots \tag{2.61}$$

and using this expansion in equation 2.58 along with equation 2.56, shows that in order to satisfy equation 2.58, the directions must obey the relation

$$\mathbf{d}^{(\tau)} \cdot \mathbf{H} \cdot \mathbf{d}^{(\tau+1)} = 0 \tag{2.62}$$

The vectors $\mathbf{d}^{(\tau)}$ and $\mathbf{d}^{(\tau+1)}$ are then said to be *conjugate.*

To make use of this relation in practice when minimising a non-quadratic function would require that the Hessian matrix be re-evaluated at each step, which is computationally expensive. It is possible to show[11] that the sequence of conjugate directions can be generated without using the Hessian matrix by making use of equation 2.57 and obtaining values for β_τ using the *Polak-Ribiere* rule

$$\beta_\tau = \frac{\left(\nabla E^{(\tau+1)} - \nabla E^{(\tau)}\right) \cdot \nabla E^{(\tau+1)}}{\left(\nabla E^{(\tau)}\right)^2} \tag{2.63}$$

There are two other possible rules for calculating β_τ, the *Hestenes-Stiefel* rule and the *Fletcher-Reeves* rule[12], which are equivalent if the function being minimised is exactly quadratic. It has been observed that the Polak-Ribiere rule performs better when minimising non-quadratic functions. On a strictly quadratic surface in n dimensions, the conjugate gradient algorithm will reach the minimum in exactly n steps, but usually requires more steps on non-quadratic surfaces.

[11] see Press et al. [66] and Bishop [3]

[12] See Bishop [3] page 280

Simulated annealing

Simulated annealing is an optimisation algorithm suitable for large-scale optimisation problems, especially those with large numbers of local minima surrounding the global minimum. It is based on an analogy with thermodynamics, which is that if a metal is heated and then allowed to cool sufficiently slowly (*anneal*), the atoms will arrange themselves in the minimum energy configuration for that system.

Metropolis et al. [55] efficiently simulated a thermodynamic system in equilibrium at a given temperature. At each step of the algorithm, an atom is given a small random displacement, changing the energy from E_1 to E_2. The probability of accepting this new configuration is given by the Boltzmann distribution

$$p = \exp\left(-\frac{E_2 - E_1}{kT}\right) \tag{2.64}$$

Note that if $E_2 \leq E_1$, then $p \geq 1$, and the new configuration is always accepted.

This algorithm can easily be applied to optimisation problems if the energy is replaced by the function which must be minimised, and the configurations are replaced by the set of parameters of the function. A random number generator must be used to make random changes to the values of the parameters. The denominator kT in equation 2.64 is replaced by a control parameter (the effective temperature) with the same units as the function.

The simulated annealing process starts with a high effective temperature, which then slowly decreases until the system "freezes" and no further changes occur. At each effective temperature, enough iterations must be done to cause the system to reach a steady state. The sequence of temperature reductions and the number of iterations to be carried out at each temperature can be considered an *annealing schedule* [37].

Masters [50] outlines how the simulated annealing algorithm can be used in training neural networks. It is not intended to be a replacement for one of the gradient descent methods, but rather a way of initialising the network weights and of attempting to break out of local minima. The standard deviation of the random number generator is used as the effective temperature. When using simulated annealing to initialise network weights, they are first set to values provided by the user (usually random values), and the standard deviation (temperature) is set to a high value. The weights are perturbed randomly n_a times. After each weight perturbation, the error function is evaluated, and if it has decreased, the values of the weights are stored and the iteration counter is set back by n_s iterations (or to zero if n_s is larger than the number of iterations performed). The purpose of setting back the iteration counter is to allow the algorithm to run for longer at a certain temperature while improvements are being made. When the number of iterations performed is greater than n_a, the weights are set to the stored best values, the temperature is reduced and the process is repeated. Masters recommends reducing the temperature by multiplying by a

constant factor each time. The factor is computed as

$$c = \exp\left(\frac{\ln\left(\frac{T_{\text{stop}}}{T_{\text{start}}}\right)}{n_t - 1}\right) \tag{2.65}$$

where T_{start} and T_{stop} are the starting and stopping temperatures, and n_t is the number of temperatures at which the procedure should run. It should be noted that this form of simulated annealing is deterministic, as only values of parameters which lead to a decrease in the error function are accepted.

2.5.6 Regularisation

An inherent disadvantage of neural networks is their large number of adjustable parameters. This corresponds to fitting a high order polynomial to a few data points, as illustrated in Figure 2.3. The technique of regularisation [3] [68] attempts to prevent over-fitting and encourage smoother network mappings by adding a penalty term Ω to the error function

$$\tilde{E} = E + \nu\Omega \tag{2.66}$$

One of the simplest forms of regularisation known as *weight decay* entails setting

$$\Omega = \frac{1}{2}\sum_i w_i^2 \tag{2.67}$$

where the sum is over all the weights in the network. This should prevent weights from saturating at large values and leading to over-fitting[13].

An indirect way of implementing regularisation is to use *early stopping*, which involves stopping the training of the neural network before it has converged. Deciding on the optimal stopping time is difficult. One way of implementing early stopping is to split the training set into a training and a validation subset. The training subset is used to train the network, while the network is tested on the validation subset regularly during the training procedure. The performance on the validation set should improve at first and reach a minimum when the network has the best generalisation ability, and then start increasing. Stopping the training when the performance on the validation set reaches a minimum should therefore result in a network with the best generalisation ability. Possible difficulties with early stopping are that the error of the network tested on the validation set does not necessarily decrease monotonically, and the values of the final weights are dependent on the values initially chosen for the weights.

Another approach to regularisation is to add a small amount of random noise to each pattern vector each time it is presented to the network during training. This is intended to prevent the network from becoming over-fitted to one set of pattern vectors.

[13] Further justification for weight decay is given in Bishop [3] pages 338–340

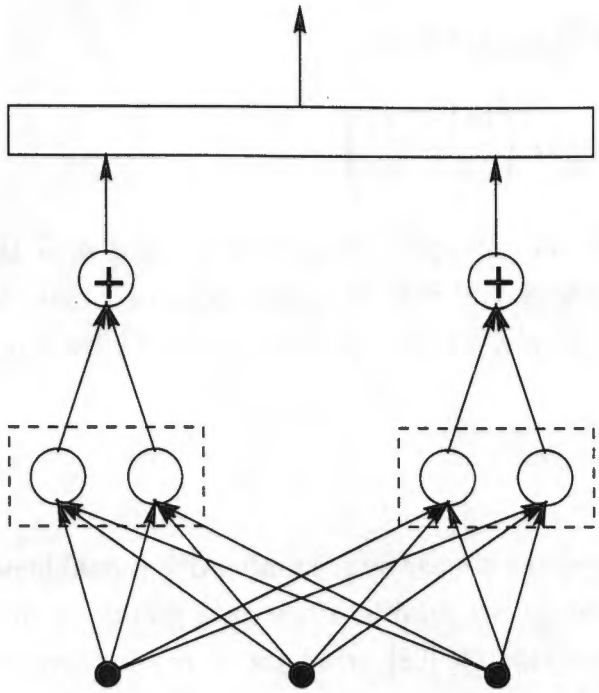


Figure 2.12: A simple probabilistic neural network. It consists of three units in the input layer, four units in the pattern layer divided into two groups or classes, and two units in the summation layer.

2.6 Probabilistic neural networks

By 1990, computers were powerful enough to allow practical use of kernel based methods (Section 2.4.1). Specht [75] revived interest in these methods by casting them into the form of a neural network, which he called a *Probabilistic Neural Network* (PNN) [50].

Figure 2.12 shows a simple probabilistic neural network. The input layer has d units, where d is the dimension of the pattern vector. The pattern layer has N units, where N is the number of training patterns. These units are grouped into K groups as shown, where K is the number of classes. Each unit in the pattern layer receives inputs from all the units in the input layer. The summation layer has K units, each of them receiving inputs from the units in one group in the pattern layer.

During training, each unit in the pattern layer is set to store one pattern $\mathbf{T}_j$ in the training set. All the units in one group contain patterns belonging to the same class. The only other task which must be done during training is to decide the value of σ to be used in equation 2.68 below.

When the network is used for classification, the unknown pattern vector $\mathbf{X}$ is presented to the input layer. This pattern vector is passed to every unit in the pattern layer. These units each calculate the value of

$$p_j(\mathbf{X}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left\{-\frac{\|\mathbf{T}_j - \mathbf{X}\|}{2\sigma^2}\right\} \tag{2.68}$$

Each unit in the summation layer then calculates the sum of all the units feeding into it and normalises the sum based on the number of inputs, effectively calculating equation 2.28 for each class. The final layer chooses the class with the largest sum and outputs that class. The advantage of this structure is that it can easily be implemented in parallel. Masters [52] presents a number of enhancements to this basic PNN, including the use of different values of σ for each variable or for each class, and a number of efficient training algorithms.

2.7 Genetic algorithms

Genetic algorithms are optimisation routines inspired by natural biological evolution. Using a genetic algorithm involves a number of iterations of a fixed-size population of candidate solutions called strings[14]. At each iteration, the members of the population interact with each other in order to improve the solution. This section presents a simple example of a genetic algorithm which will assist in understanding the genetic algorithm applied to chromosome classification in Section 4.4.6. More details on genetic algorithms can be found in Davis [8], Goldberg [16] or Grefenstette [23].

The simple genetic algorithm described here can be applied to any system or function which takes a binary vector as an input and produces a scalar output. Assume that the algorithm will be used to minimise an arbitrary function $E(\mathbf{x})$, where $\mathbf{x}$ is a binary vector. To start the algorithm, members of an initial population P_0 of strings, where each string is a possible value of $\mathbf{x}$, are constructed randomly. The population at the next iteration P_1 is then constructed by using two operators — *one-point crossover* and *mutation* — on population P_0. Two different strings in population P_0 are chosen using a random procedure which gives larger weighting to strings which produce smaller values of $E(\mathbf{x})$. These strings are copied into population P_1 after which they are operated on by the one-point crossover operator, which chooses a random position on the string, and exchanges string segments to the right of the chosen position. For example, if the two strings undergoing crossover are represented by 11101:001 and 10110:111, where the colon marks the crossover position, then the two resulting strings will be 11101:111 and 10110:001. These strings are then operated on by the mutation operator, which inverts a few randomly chosen bits in the strings. Once the number of strings in P_1 is equal to the number of string in P_0, then the process is repeated to construct P_2 from P_1, etc. The procedure can be stopped once a fixed number of populations have been generated, or if the value of $E(\mathbf{x})$ gets small enough. At the end of the procedure, the string which produces the lowest value of $E(\mathbf{x})$ is chosen as the solution vector.

In order to use this algorithm on real optimisation problems, a method of encoding solution vectors containing real numbers to binary form must be implemented, or other

[14]These candidate solutions are also sometimes called chromosomes in analogy to the biological structures which they are simulating.

forms of the operators which operate on real valued strings must be used.

2.8 Measuring the performance of classifiers

The parameter used most often to quantify the performance of a classifier is the number of misclassified patterns in the test set divided by the total number of patterns in the test set (often quoted as a percentage).

Testing a classifier using the same data used to train it can lead to over-optimistic error estimate [9]. The *hold out* method was of the earliest proposed solutions to this problem. It involves dividing the training set into two parts, training the classifier on one part and testing it on the other. This approach leads to part of the training data being wasted in that it is not included in the training of the classifier.

Cross-validation makes better use of the training data. The simplest form of cross validation involves splitting the training set into two halves, A and B. The classifier is trained on part A and an error estimate is obtained by testing the classifier on part B. The roles of the two halves are then swapped, and the classifier is retrained on part B and tested on part A. An unbiased estimate of the classification error is obtained by averaging the two classification errors [68]. If the amount of available training data is small so that keeping half of it aside leads to poor performance of the classifier, it is possible to divide the training set into V parts. The classifier will then have to be trained V times, each time using $(V-1)$ parts of the training set to train the classifier and testing the classifier on the remaining part. The error is estimated by averaging the V classification errors. If the training set is divided so that V is equal to the number of patterns in the set, this leads to the *leave-one-out* method.

Other methods such as the *jackknife* and *bootstrap* methods[15] attempt to provide estimates of the error using the same set for training and testing. The jackknife method is often confused with leave-one-out cross-validation.

[15] See Ripley [68] pages 72–75

Chapter 3

Extracting Features from Chromosomes

A digitised chromosome image consists of a two-dimensional array of grey pixel values. In order to carry out automatic classification, a collection of distinguishing features must be extracted from each chromosome. These can include area, length, centromere position, one-dimensional banding pattern and information about each band. This chapter describes the various techniques that have been used to extract chromosome features. These techniques all assume that the metaphase image has been thresholded to separate the chromosomes from the background, and that all touching and overlapping chromosomes have been separated so that each chromosome is marked as a separate object. The automatic separation of touching and overlapping chromosomes is a difficult problem which will not be covered here.

3.1 Global features

Once each chromosome is marked, it is simple to measure the area and density (sum of the pixel grey-values) of each chromosome. The convex hull can also be extracted at this stage, and the minimum width enclosing rectangle can be used to orient the chromosome vertically (Figure 3.1).

3.2 Integrated density profile

The bands are usually perpendicular to the chromosome axis. It is therefore possible to represent the bands by a one-dimensional integrated density profile. A schematic diagram demonstrating how the integrated density profile is constructed is shown in Figure 3.2.

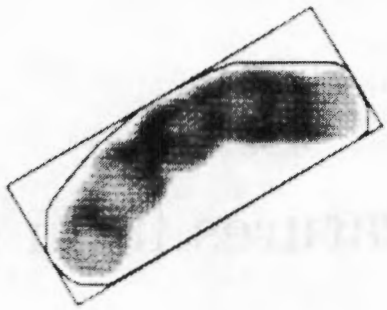


Figure 3.1: A chromosome, its convex hull, and the minimum width enclosing rectangle (from [65]).

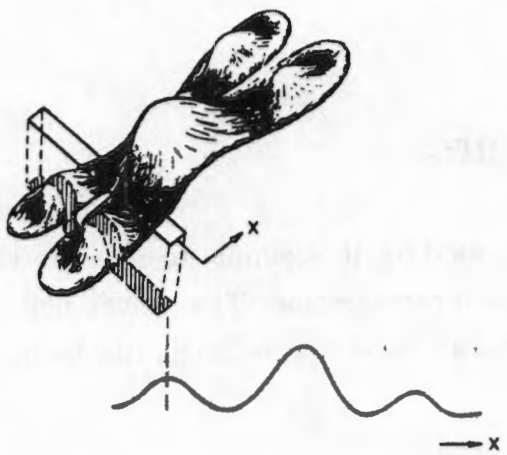


Figure 3.2: Schematic diagram showing how an integrated density profile is generated (from [20]).

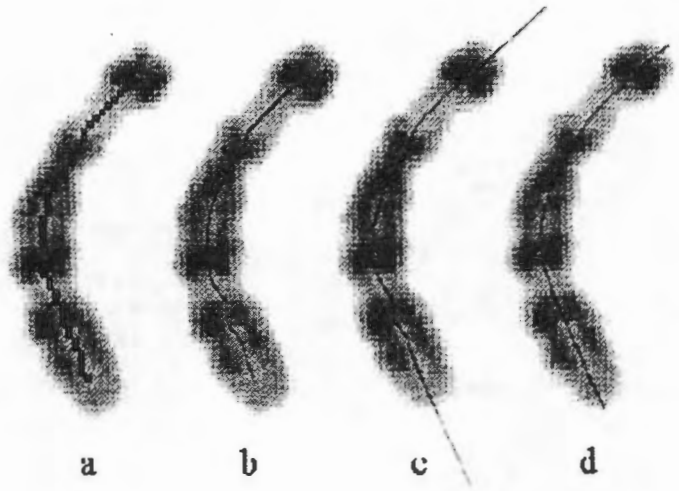


Figure 3.3: Finding the axis of a chromosome using skeletonisation. **a:** The skeleton of the chromosome is computed. **b:** The skeleton is represented by a polygon. **c:** The tips of the polygon are extended. **d:** The polygon is smoothed and the tips are restricted to the chromosome body (from [65]).

3.2.1 Finding the chromosome axis

Before an integrated density profile can be extracted, the chromosome axis has to be found. A number of parametric curves have been proposed and tried as a fit to the chromosome axis [59]. Two of the more recent axis-finding algorithms are described below.

Piper and Granum [65] propose the method illustrated in Figure 3.3. The chromosome axis is located using a skeletonisation algorithm[1] (Figure 3.3a) and a polygon is fitted to the skeleton (Figure 3.3b). As the skeleton does not extend to the tips of the chromosome, the ends of the polygon are extended (Figure 3.3c). Finally, the polygon is smoothed by convolution with a low-pass filter. Piper and Granum [65] also outline a "poor man's skeleton" algorithm which can be used to find the axis of straight or slightly bent chromosomes in less time than conventional skeletonisation.

Groen et al. [24] proposed using a piecewise-linear (PWL) approximation to the axis. A rough approximation to the principal axis is calculated and the middle axis of the chromosome is found by taking lines perpendicular to this axis (Figure 3.4a). As the lines are perpendicular to the principal axis and not to the chromosome, artifacts sometimes occur at the tips of the chromosome, so the tips are ignored in the initial stages of the PWL calculation. An iterative procedure is then used to find the PWL axis. A line is drawn joining the end points of the middle axis, and the point on the middle axis furthest away from this line is found. Two lines are then drawn from this furthest point to the middle axis end-points. This procedure is continued until all middle axis points are within a certain distance of the PWL approximation (Figure 3.4b). As lines perpendicular to the PWL

[1]See a standard image processing text such as Jain [31] pages 381–390

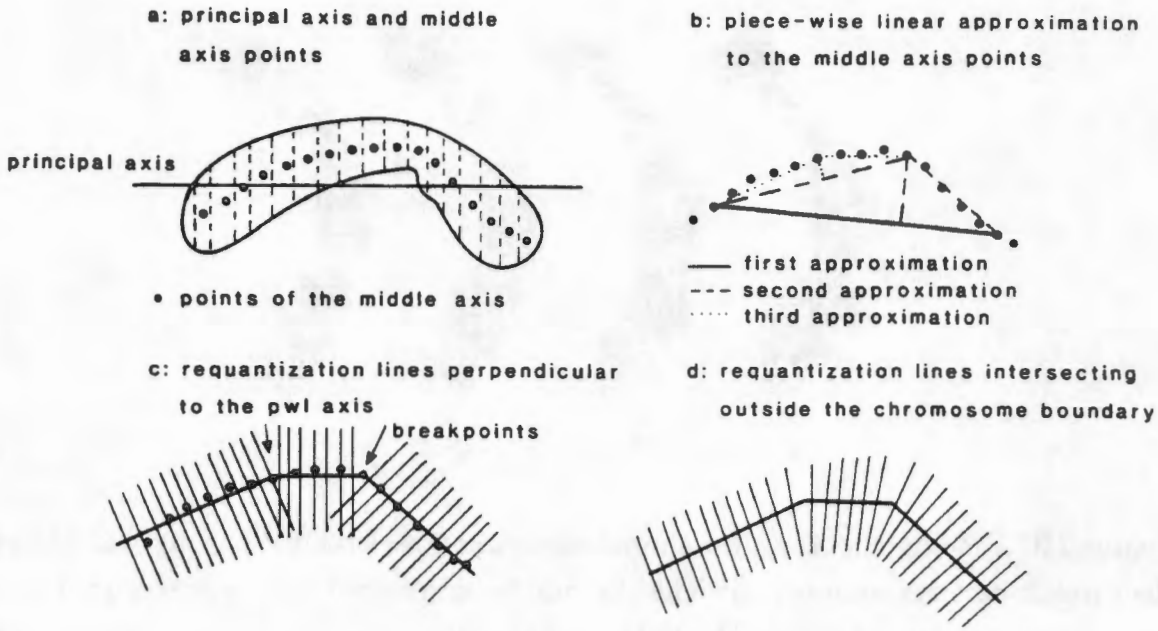


Figure 3.4: Construction of the piecewise-linear (PWL) axis. **a:** The middle of the chromosome is computed in columns perpendicular to the principal axis. **b:** A piecewise-linear approximation is fitted to these middle axis points. **c:** Lines perpendicular to the axis intersect inside the chromosome. **d:** The lines are therefore adjusted near the breakpoints so that they do not intersect (from [24]).

approximation intersect inside the chromosome (Figure 3.4c), some adjustments are made to these lines around the breakpoints of the PWL axis (Figure 3.4d).

3.2.2 Calculating the density profile

The density values (or grey-levels) are summed along each line perpendicular to the axis to get the integrated density profile. A fixed distance is chosen between all the sampled points on the transverse lines, and the grey-levels of points which do not fall on pixels are obtained by two-dimensional interpolation. The upper profiles in Figure 3.5 are integrated density profiles.

3.2.3 Possible limitations of the integrated density profile

Groen et al. [24] pointed out some possible limitations in using the integrated density profile to represent the banding pattern. It is possible that the chromatids can contract unequally and lead to the smearing of bands in the profile as illustrated in Figure 3.6a. Bands can also deviate from a rectangular shape resulting in the situation shown in Figure 3.6b.

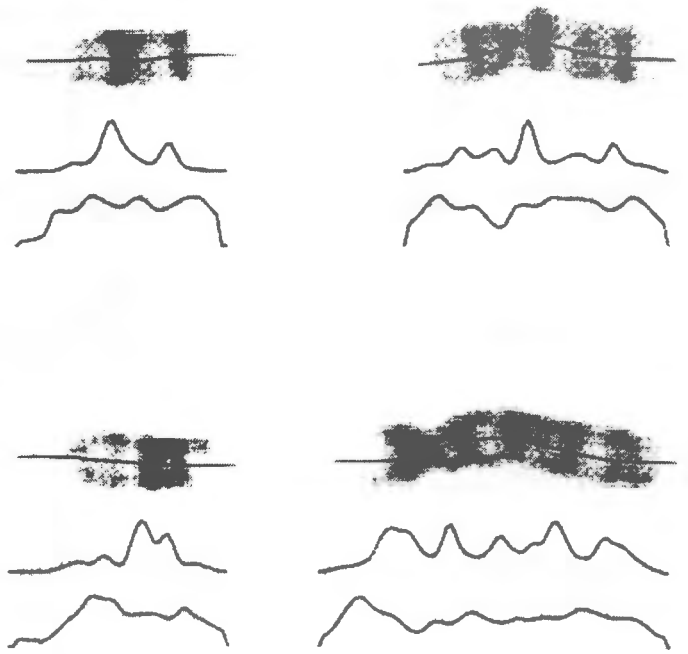


Figure 3.5: Banded chromosomes with their integrated density (upper) and "shape" (lower) profiles. The "shape" profiles are used to locate the centromeres (from [65]).

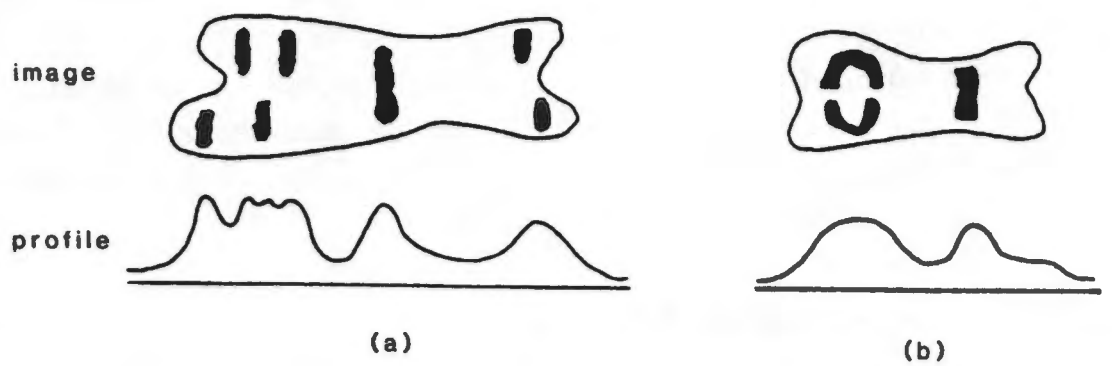


Figure 3.6: Potential problems with using an integrated density profile to describe a chromosome banding pattern. **a:** Unequal contraction of chromatids leads to smearing of bands. **b:** Protrusions may results in artificial fusion of bands (from [24]).

Metacentric Chromosome Acrocentric Chromosome

Figure 3.7: Centromere finding with an exhaustive search for the closest pair of opposite edge points. The points are found at the end of the chromosome for acrocentric chromosomes (from [24])

3.3 Locating the centromere

Piper [60] pointed out that there are two basic types of centromere finding algorithms, boundary analysis algorithms and algorithms which analyse density profiles. Three algorithms for locating centromeres described relatively recently are presented below.

3.3.1 Boundary analysis

Groen et al. [24] proposed two methods of finding the chromosome centromere using boundary analysis. The first method, illustrated in Figure 3.7, is based on searching for the closest pair of opposite edge points. The head and tail of the chromosome are deleted, with the amount deleted chosen to be equal to the size of the p-arm terminal in an acrocentric chromosome. The centromere is then assumed to be at the position of the closest opposite edge points. When the two points are at the end of a chromosome, then the chromosome is assumed to be acrocentric.

In the second method, a width profile is constructed based on the lengths of lines perpendicular to the fitted chromosome axis. The profile is smoothed and a relative minima between two maxima is searched for. If no such local minimum is found, the chromosome is assumed to be acrocentric.

3.3.2 Density profile analysis

Piper and Granum [65] find the centromere using a "shape" profile, which is computed along lines perpendicular to the chromosome axis in a similar way to the integrated density profile, and is independent of the banding pattern [60]. Each value of the shape profile S_i is the ratio of the second to the zeroth grey moment of the transverse slice, or

$$S_i = \frac{\sum_j m_j d_j^2}{\sum_j m_j} \tag{3.1}$$

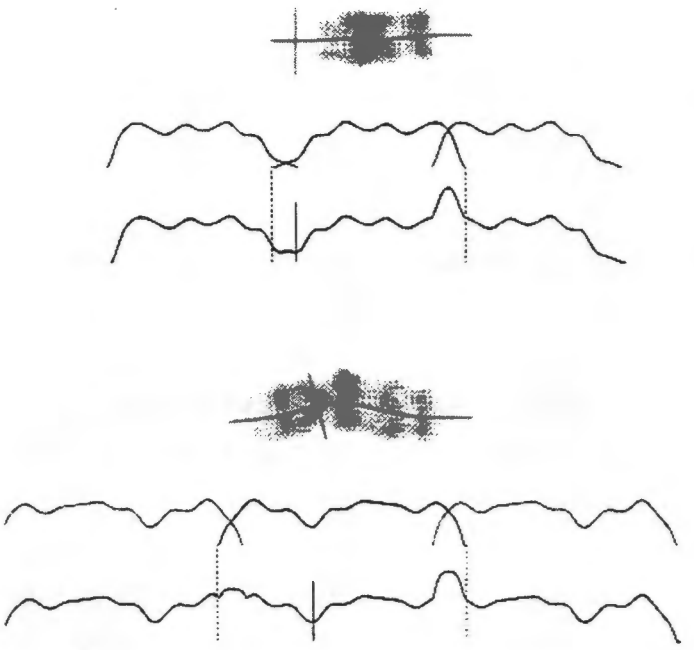


Figure 3.8: Transformation of the shape profile for detecting acrocentric centromeres. The centromere is chosen to be at the minimum of the transformed shape profile found between the dotted lines. The positions of the detected centromeres are marked on the profiles and on the images (from [65]).

where $1 \leq i \leq l$, l is the profile length, m_j is the density (grey-level) of the jth pixel on perpendicular line i, and d_j is its distance from the axis (see the lower profiles in Figure 3.5).

The centromere of a metacentric chromosome is usually found at the most pronounced minimum of the shape profile. No such minimum is found on the shape profile of an acrocentric chromosome, and a minimum has to be induced by reflecting the shape profile close to each end. If the shape profile S_i has non-zero values in the range 0 to l and is zero elsewhere, then the transformed profile is

$$S'_i = S_i + S_{k-i} + S_{2l-k-i} \tag{3.2}$$

where $k < l$ is the length of overlap between the original profile and its reflections as shown in Figure 3.8. This transformation tends to induce a minimum at a distance of about $\frac{k}{2}$ from the centromere end of acrocentric chromosomes, so k is chosen such that $\frac{k}{2l}$ is the average centromeric index for acrocentric chromosomes. If minima are chosen within the bounds of the original profile ($0 \leq i \leq l$), then the centromere positions for metacentric chromosomes are not usually affected.

3.3.3 Comparison of the methods

Table 3.1 shows a comparison of the accuracy of the three centromere finding algorithms described above. Piper and Granum [65] defined a correct centromere as one which lies

Authors	Method	Percentage of correct centromere positions found
Groen et al. [24]	closest pair of opposite edge points	93
Groen et al. [24]	local minimum of width profile	76
Piper and Granum [65]	"shape" profile	93.5

Table 3.1: Comparison of the performance of the three centromere finding techniques described in the text. All the tests were performed on the Copenhagen database with overlapped chromosomes excluded. The database used by Groen et al. [24] contained 7284 chromosomes and the database used by Piper and Granum [65] contained 8106 chromosomes. See the text for details on the slight differences in the definition of a correctly positioned centromere used by the authors.

within two pixels of the manually determined centromere for metacentric chromosomes, and either satisfies the previous criterion or lies between the manually located centromere and tip of the short arm in acrocentric chromosomes. Groen et al. [24] considered an automatically determined centromere to be correct if the distance between it and the manually located centromere was less than 10% of the chromosome length.

3.4 Features derived from the profile

3.4.1 Band transition sequences

A Band Transition Sequence (BT-sequence) [47] is constructed from an integrated density profile as shown in Figure 3.9. Part **B** of Figure 3.9 is constructed from part **A** by the application of an iterative non-linear filter. The length of the chromosome (x-axis) is divided into 13 segments. The BT-sequence (part **C**) consists of fourteen BT-codes, one code corresponding to each length segment and one code for the terminal end of the short arm. Each segment code is constructed by determining the 'peak density' and 'density difference between peak and adjacent valley' in terms of density classes for the peak in the segment. The 'peak density' is represented by one of six density classes and the 'density difference' by one of four difference classes. If a segment contains no peak, then the 'peak density' and 'density difference' are assigned to class 0. These BT-codes are excluded from the print-out in Figure 3.9.

3.4.2 Weighted density distributions

Chromosome density profiles can be described by weighted density distributions (wdd's) by the application of a number of weighting functions [46]. The weighting functions were initially defined by Granum [21]. Figure 3.10 illustrates the following equation used to

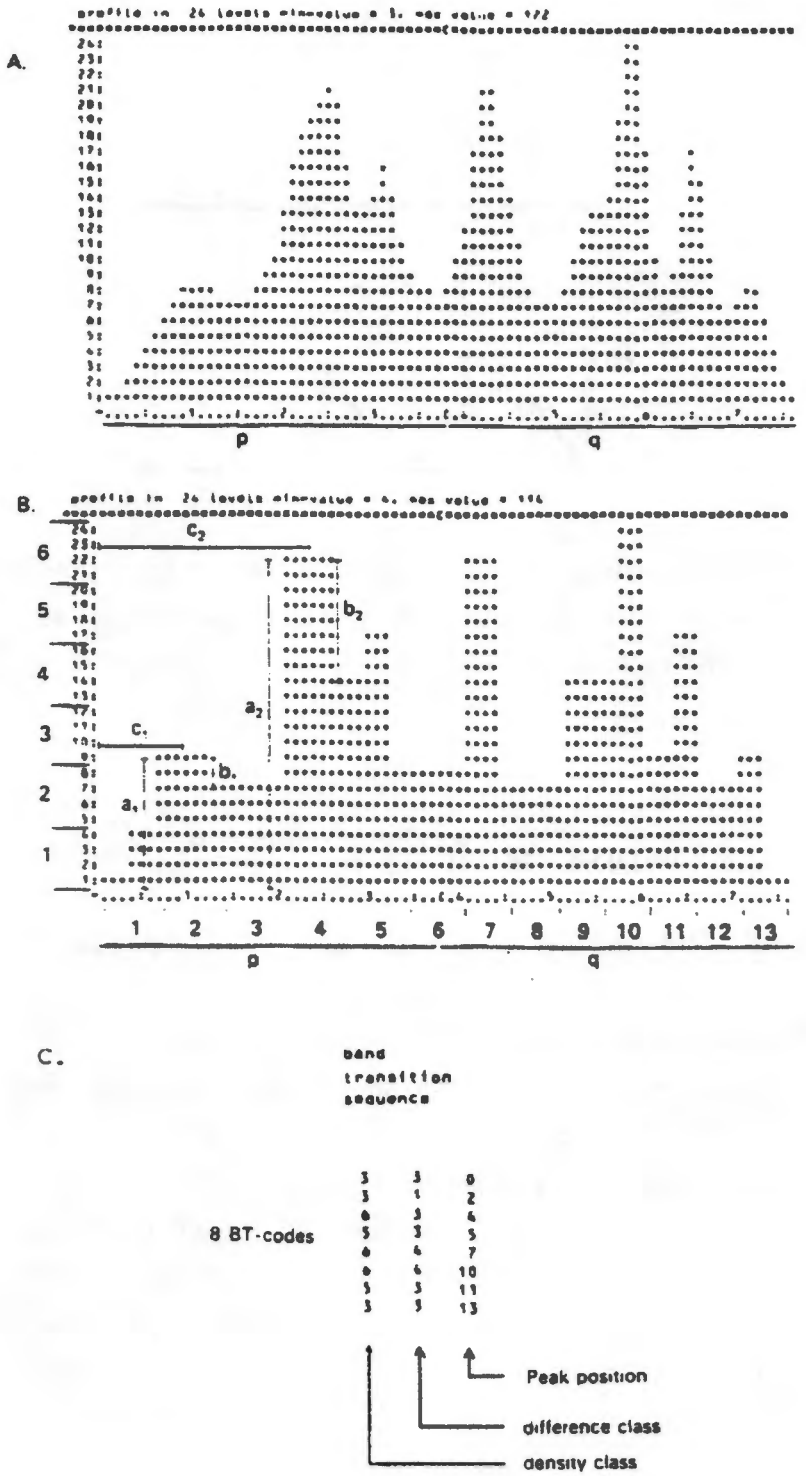


Figure 3.9: Construction of a BT-sequence from a chromosome integrated density profile. **A** is the integrated density profile of a chromosome of class 1 printed out in 24 density levels. **B** is a sharp-edged profile of the same chromosome. **a**, **b** and **c** refer to peak density **(a)**, density difference between peak and adjacent valley **(b)** and peak position **(c)**. **C** shows the BT-sequence (comprising 8 BT-codes) constructed from **B**. See the text for more details (from [49]).

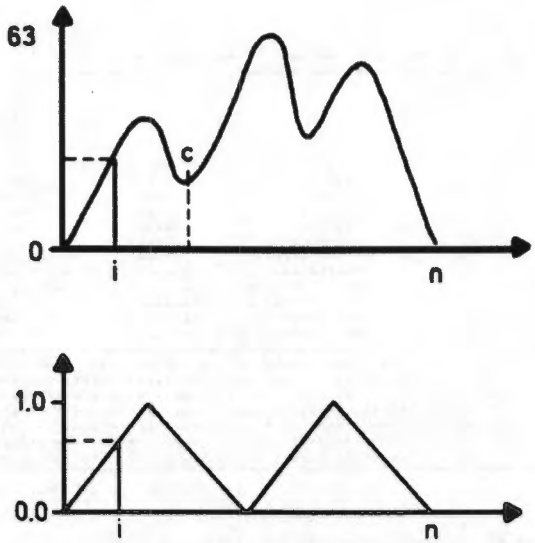


Figure 3.10: Illustration of a chromosome profile weighted by a weighting function. The sum ($i = 1$ to n) of all products of density values and corresponding weight factors is divided by the integrated density of the profile. c indicates the position of the centromere (from [46]).

calculate a wdd:

$$\text{WDD} = \frac{\sum_{i=0}^{n} \text{weighting factor}_i \times \text{density}_i}{\sum_{i=0}^{n} \text{density}_i} \tag{3.3}$$

Lundsteen et al. [46] and Gerdes and Lundsteen [14] expanded the original Granum weighting functions to those shown in Figure 3.11.

Piper and Granum [65] also made use of weighted density distributions to describe chromosomes, although they redefined the weighting functions as shown in Figure 3.12. The redefined weighting functions are either symmetrical (wdd 2, 4, 6) or asymmetrical (wdd 1, 2p, 3), have integrals of zero and are independent of the centromere position, although the "asymmetrical" weighting functions do depend on the chromosome polarity. These weighting functions are applied to the integrated density profiles to extract six wdd features (wdd1, wdd2, wdd2p, wdd3, wdd4 and wdd6); to the profiles of absolute differences of the density profile P, defined as $G(i) = |P(i) - P(i-1)|$, to extract six gwdd features; and to the "shape" profiles (Figure 3.5) to extract six mwdd features.

3.5 Other features

Other features defined by Granum [21] and Piper and Granum [65] are:

- Relative Density, defined as the ratio of total optical density to area.
- Convex Hull Perimeter (cvhp) (see Figure 3.1).

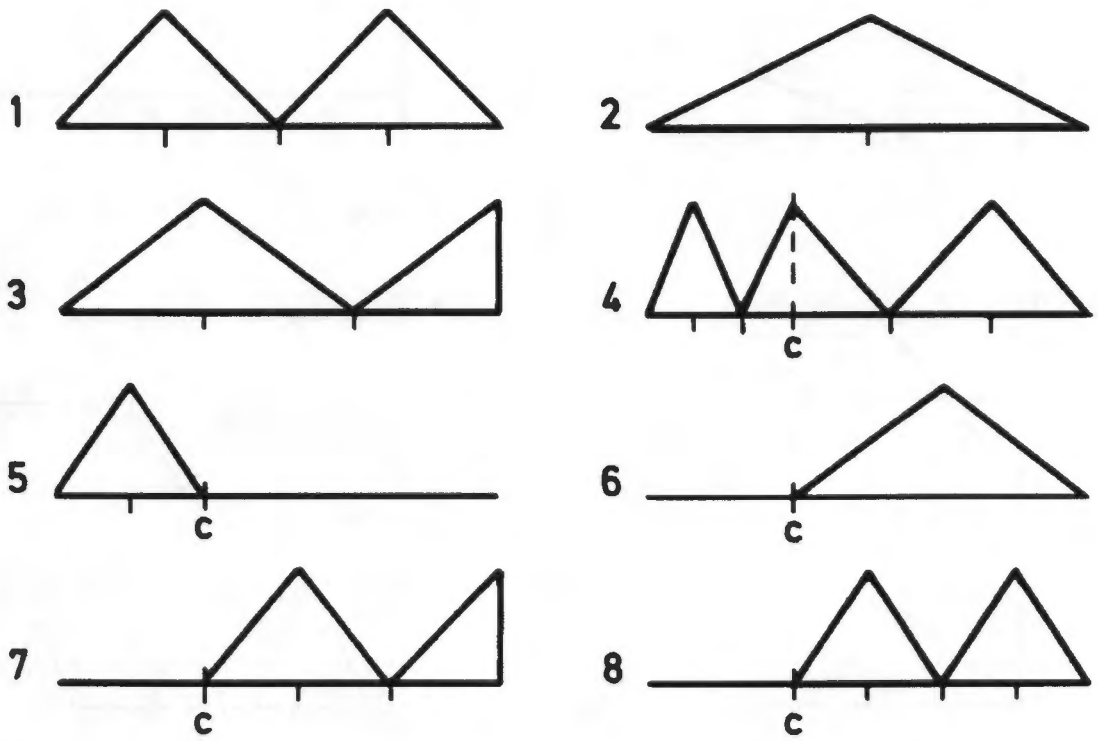


Figure 3.11: Illustration of the eight weighting functions used by Lundsteen et al. [46], which are used to compute the corresponding eight wdd's on the basis of a chromosome density profile. c indicates the position of the centromere. The first three weighting functions are independent of the position of the centromere, but this is taken into account in the remaining five. The weighting functions are scaled to match the length and centromere position of each chromosome.

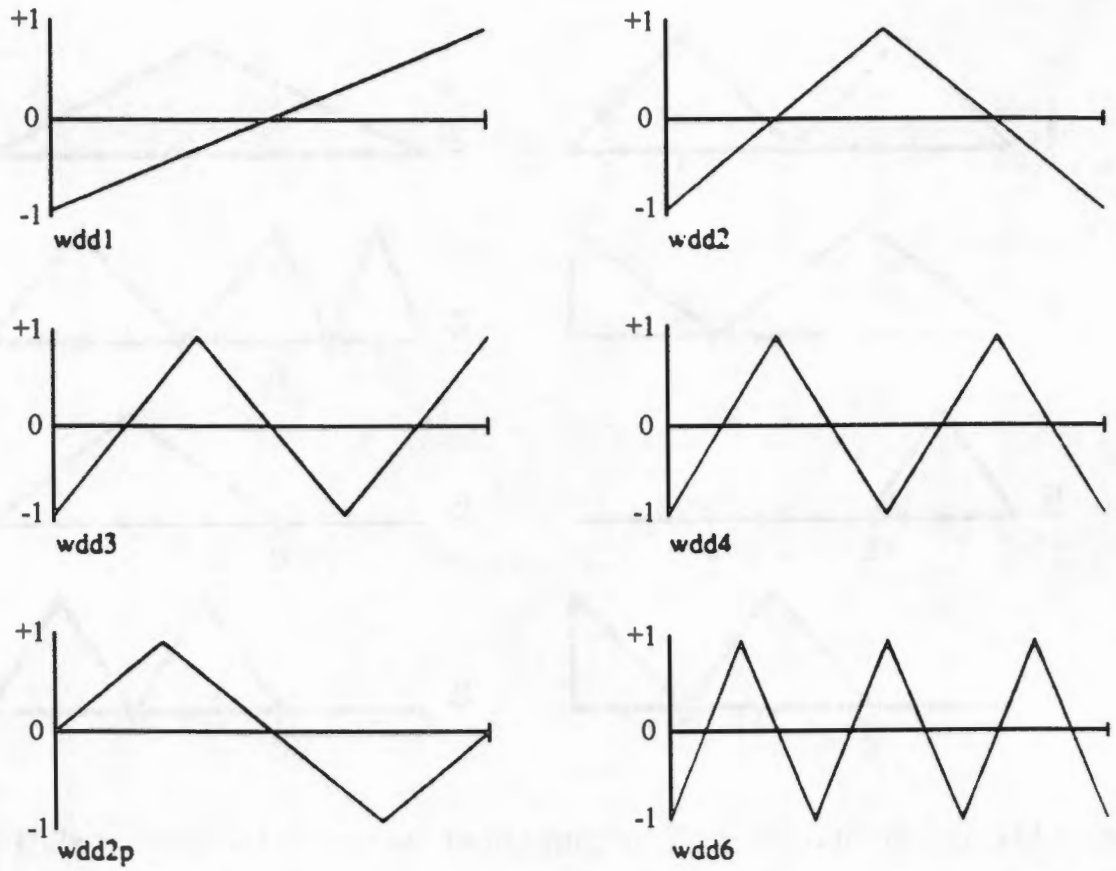


Figure 3.12: The wdd functions used by Piper and Granum [65]. Wdd 2, 4 and 6 are symmetrical and hence are independent of the chromosome polarity. Wdd 1, 2p and 3 are asymmetrical and so are affected by the chromosome polarity. The integrals of these functions are zero and they are independent of the centromere position.

- Size, defined as the mean of the normalised area and normalised length.
- Coefficient of variation of the density distribution (cvdd), defined as the ratio of the standard deviation to the mean of the values of the density profile.
- Normalised root of the sum of squared density differences (nssd), defined as

$$\text{nssd} = k\frac{\sqrt{\sum_{i=1}^{n}\left(P\left(i\right)-P\left(i-1\right)\right)^{2}}}{\sum_{i=0}^{n}P\left(i\right)}$$

  where k is a scaling factor.
- Number of density profile maxima (number of bands) (nb)
- Number of bands index (nbi), defined as the ratio of the number of bands in half the profile to the total number of bands in the profile.

3.6 Feature levels

Granum [21] and Piper and Granum [65] divide the features used to describe chromosomes into four levels. The level 1 features can be measured directly from a chromosome image and include the area and relative density. The axis needs to be found in order to compute level 2 features, which include the length, density profile and "symmetrical" wdd and gwdd features. Level 3 features require the axis profile and chromosome polarity to be known, and include the "asymmetrical" wdd and gwdd features. Level 4 features are the centromeric indices which require all the level 3 features and the position of the centromere to be known. Table 3.2 shows the features defined above and the corresponding feature levels. The band transition sequence features are level 3 features.

3.7 Feature normalisation

Chromosomes in different metaphases contract by different amounts and can have different staining intensities. For example, chromosomes of the same type may vary in size by a factor of 2–3 between different metaphases [6]. Hence, in order to be able to compare cells in different metaphases to each other, the feature measurements, most notably the lengths, have to be normalised. This is easy to do if the classes of the chromosomes are known, but difficult to do before the chromosomes have been classified. Hilditch and Rutovitz [28] and Moore [30] proposed iterative methods which initially make a rough guess at a normalisation factor, tentatively classify some chromosomes, refine the normalisation factor based on the classified chromosomes, classify some more chromosomes, and continue until the normalisation factor converges.

Feature	Level	Feature	Level
Area	1	gwdd2p	3
Relative Density	1	gwdd3	3
cvhp	1	gwdd4	2
Length	2	gwdd6	2
Size	2	mwdd1	3
cvdd	2	mwdd2	2
nssd	2	mwdd2p	3
wdd1	3	mwdd3	3
wdd2	2	mwdd4	2
wdd2p	3	mwdd6	2
wdd3	3	Area c.i.	4
wdd4	2	Density c.i.	4
wdd6	2	Length c.i.	4
gwdd1	3	nb	2
gwdd2	2	nbi	3

Table 3.2: The feature levels of all the chromosome features defined in the text. Partly from [65].

Piper and Granum [65] used a simpler normalisation method for their features (Table 3.2). Size-type features are normalised to the cell median measurement for that feature. The median is chosen instead of the alternatives — the mean or the sum of the feature measurement over all the chromosomes in the cell — as it is less sensitive to missing or additional chromosomes or undetected composites of two or more chromosomes. For other features, the measurements within each cell are normalised to have a mean of zero and a standard deviation of 100. The centromeric index is expressed as a proportion of chromosome length, and so is automatically normalised for metaphase to metaphase length variation.

3.8 Feature selection

As using a large number of features can lead to over-fitting of the model to the training data and to long computation times when classification is performed, it is often advantageous to reduce the number of features by keeping the features with the largest discriminatory ability. An exhaustive search through all the possible feature combinations is impractical, so sub-optimum methods must be used. Granum [21] tested a number of standard statistical methods, but found that an heuristic method which he called SEPCOR outperformed them.

Granum calculates a separability measure v_i for each feature, where

$$v_i = \frac{\text{standard deviation}\,\{\mu_{ik} | k = 1, 2, \ldots, m\}}{\text{mean}\,\{\sigma_{ik} | k = 1, 2, \ldots, m\}}$$

and μ_{ik} and σ_{ik} are respectively the mean and standard deviation of the value of feature i for a chromosome of true class k out of m classes. The correlation coefficient between two features i and j is r_{ij}. The SEPCOR procedure attempts to assign a specified number of features from a list of all the features to a 'selected' subset and the rest to a 'discarded' subset. The following steps are carried out:

1. The feature with the maximum v_i is chosen from the list and added to the 'selected' subset.

2. All features in the 'selected' subset are correlated with features remaining in the list, and those features in the list with correlation coefficients greater than a parameter MAXCOR are moved to the 'discarded' subset.

3. The process is continued until the required number of features has been selected or the list is empty.

The disadvantages of this method, namely the heuristic parameter MAXCOR and the tendency for the list to empty before enough features have been selected, are addressed by Piper's MSEPCOR method [62]. The steps in the MSEPCOR procedure are:

1. The feature with the maximum v_i is chosen from the list and added to the 'selected' subset.

2. The maximum correlations $R_u = \max_s \{|r_{us}|\}$ between each unselected feature u in the list and all 'selected' features s are calculated.

3. The feature u for which $v_u\,(1 - R_u)$ is a maximum is added to the 'selected' subset.

4. Steps 2 and 3 are repeated until the required number of features are selected.

The results of running the MSEPCOR procedure on the Cph, Edi and Phi data sets to choose subsets of 10 or 16 features from those listed in Table 3.2 are shown in Table 3.3 [65]. Feature sets were chosen for each data set separately and a pooled feature set using all three data sets was chosen.

Feature	Derivation of feature subsets			
	Pooled among data sets	Within each data set		
		Cph	Edi	Phi
Area				
Relative Density				+
cvhp				
Length				++
Size	++	++	++	
cvdd			+	
nssd	+	+	++	++
wdd1				
wdd2	++		++	++
wdd2p	++	++	++	
wdd3	++	++	++	++
wdd4	++	++	++	++
wdd6	++	++	++	++
gwdd1	+	+	+	+
gwdd2	++	++	++	++
gwdd2p	+	+		+
gwdd3	+	+	++	+
gwdd4		+	+	+
gwdd6	+	++	+	
mwdd1				
mwdd2	+	+		
mwdd2p				
mwdd3				+
mwdd4	++	++	+	++
mwdd6	++	++	+	++
Area c.i.	++		++	
Density c.i.		++		++
Length c.i.				

Table 3.3: Feature subsets consisting of 10 and 16 features constructed using the MSEPCOR method from the pooled data sets and from each data set individually. Features in the size 10 subsets are marked "++". All these features are also in the corresponding size 16 subsets, and the additional six features in each subset are marked "+" (from [65]).

Chapter 4

Automatic Classification of Chromosomes

This chapter reviews the algorithms which have been applied to chromosome classification and shows experimental results of the algorithms applied to some of the data sets described in Section 1.5. The chief differences between the algorithms are in the classifier used and the features passed to the classifier. Feature extraction methods are treated in detail in the previous chapter. The classifiers described in this chapter are divided into two groups based on the types of features passed to the classifier. The features can be

1. A direct representation of the sampled chromosome profile, possibly with extra information on the length and centromeric index (Section 4.3).
2. the set of features listed in Table 3.2 extracted from each chromosome (Section 4.4).

Further topics covered are the advantages and methods of taking the number of chromosomes per class into account (Section 4.1), the reliability of chromosome classification by humans (Section 4.2) and a comparison of the results produced by the best classification algorithms reported in the literature (Section 4.5).

4.1 Taking the number of chromosomes per class into account

The simplest method of classifying a chromosome is simply to calculate the probability of the chromosome belonging to each class, and to assign the chromosome to the class to which it has the highest probability of belonging. Habbema [25] [26] and Slot [74] pointed out that in the case of assigning all the chromosomes in a cell to classes, it is known a priori how many chromosomes are in each class. The simplest way to take this into account is to calculate the class-membership probabilities for each chromosome (as described above),

and then include a post-processing step in which the chromosomes are moved between classes until the limit on the number of chromosomes per class is satisfied. Some of these post-processing algorithms are described in Section 4.1.1. Unfortunately, the best ways of taking the number of chromosomes per class into account are prohibitively calculation intensive. For example, using the Bayes method which gives the solution with the smallest mean number of misallocations described by Slot [74] requires the evaluation of a sum consisting of the order of 10^{55} terms when applied to chromosome classification. Habbema [26] proposes using all the chromosomes in a cell simultaneously as a basis for discriminant analysis. In practice this involves carrying out calculations on a $46 \times p$ covariance matrix, where p is the number of features extracted from each chromosome. Due to the large number of possible permutations of chromosomes in the karyotype and the large number of cells that would be needed to train such a large model without overfitting, this method is difficult to implement.

4.1.1 Chromosome rearrangement algorithms

Two rearrangement methods which are used in a number of experiments described in this chapter are outlined below. The "simple exchange" or SE algorithm is described by Lundsteen et al. [45]. Piper [61] describes and presents results of tests of four rearrangement routines which he calls RC1–4. He discovered that the RC3 method (described below) had the best performance and resulted in an improvement of between 2.2% and 4.0% when applied to likelihoods computed using varying numbers of features extracted from the Edinburgh data set and passed to the maximum likelihood routine described in Section 2.3.

SE: If more than two chromosomes are assigned to a class, only the two with the highest likelihood of belonging to the class are accepted. Any excess chromosomes are assigned to the classes for which they have the next highest likelihoods. Chromosomes which cannot be assigned to a class are rejected.

RC3: This is a modification of the algorithm proposed by Rutovitz [73]. The rearrangement is implemented as a cascade through a set of classes $G_1, G_2, \ldots, G_n$, where class G_1 has an apparent excess of chromosomes and class G_n has an apparent deficit. In each stage of the cascade, one chromosome is moved from class G_i to class G_{i+1}. The cost of moving a chromosome from class G_i to G_{i+1} is defined as $C_i = \frac{L_i}{L_{i+1}}$, where L_i is the likelihood that a chromosome is of class G_i [61]. The cost of the whole cascade is defined as $CC = \prod C_i$. Plausibility constraints allow the classifier to cope with unusual cases where, for example, there really are three chromosomes in a class (e.g. Down's syndrome). The plausibility constraints implemented by Piper [61] are:
(i) the reassignment of a chromosome to class G_j is only permitted if $L_j > k \cdot \max(L_i)$, where k is determined by experiment, and was set to 0.15 by Piper, and $\max(L_i)$ is

Experiment Number	Percentage Classification Error	
	Context-free	Context-sensitive
1	3.1	0.11
2	5.4	0.50
3	6.4	0.72

Table 4.1: The results of the three experiments described in the text, measuring the errors made when chromosomes are classified by humans. Percentage errors for both context-free and context sensitive classification results are shown.

the maximum likelihood for the chromosome.
(ii) the length of a cascade should not exceed 4 classes (3 moves).
The rearrangement procedure computes the cost of all plausible cascades and then carries out the minimum cost cascade. This procedure is iterated until no further plausible cascades can be found.

Other methods of applying constraints on the number of chromosomes per class have been proposed. The transportation method is the application of a linear programming algorithm to the problem, and is described in Section 4.4.4. The application of genetic algorithms is described in Section 4.4.6.

4.2 Chromosome classification by humans

Granum [21] quotes the results of three experiments which measured the error rate when chromosomes were classified by humans. The same data, consisting of chromosomes from 22 normal cells, were used in each experiment. Chromosomes were classified in isolation (context-free classification) or, on separate occasions, simultaneously with all other chromosomes of the same cell (context-sensitive classification or karyotyping).

In the first experiment, seven independent investigators classified chromosomes represented by photographic prints. For the second experiment, one investigator classified isolated digitised chromosome profiles in four runs at weekly intervals, and karyotyped them twice. Overlapped and bent chromosomes were excluded, which resulted in incomplete cells which are more difficult to karyotype. The third experiment was carried out in the same way as the second, except that the chromosomes were represented by band transition sequences (Section 3.4.1). The results of the three experiments are shown in Table 4.1. It is immediately obvious that a trained person classifying chromosomes uses the constraints on the number of chromosomes per class to decrease the misclassification rate significantly.

4.3 Classifiers which use sampled profiles

4.3.1 Early algorithms

One of the earlier attempts at classifying G-banded chromosomes by Granlund [20] involves extracting integrated density profiles (Section 3.2) and using some standard spectroscopic techniques to extract features from the profiles.

Techniques Used

Curve-Matching: A reference profile for each class i of chromosome is obtained by calculating the mean m_{ij} and variance s_{ij}^2 at each point j of all the profiles belonging to class i in the training set. An unknown profile x_j is compared with all the reference profiles by computing

$$\delta_i^2 = \sum_j \frac{k_{ij}\,(m_{ij} - x_j)^2}{s_{ij}^2} \tag{4.1}$$

for each class i, where k_{ij} is a weighting factor which is used to specify the relative importance of different features.

Curve-Matching with Non-Uniform Sampling: Chromosomes are initially classified using the curve-matching technique, and the two most likely classes are considered further. To make the decision between these two classes, points on the profiles with the greatest discriminatory ability are manually chosen and used to decide between the two classes.

Fourier Descriptors: Fourier coefficients are calculated for each chromosome profile in the training set and the magnitudes of harmonics 1 to 8 and the phase angles of harmonics 1 to 4 are used to calculate the means and covariance matrices for the classes. Unknown chromosomes are classified using a parametric classifier (Section 2.3).

Distribution Functions: The profile is approximated by a collection of Gaussian-like distribution functions [18] [19], as shown in Figure 4.1. The profile can then be described as a series of triplets, each including peak height, width, and position of a distribution function. Granlund postulated that using an average of the distribution function profile parameters as a reference description of a banding pattern is superior to using averages of the profiles or averages of Fourier components of the profile, as this approach is closely related to the physical banding mechanism and hence results in less "smearing" of the information. Unfortunately, as mentioned by Habbema [26], this approach does not work well in practice due to the uncertainty in the number of Gaussian curves that should be used to describe each type of chromosome and the difficulty of numerically fitting a group of non-linear functions to the profiles.
Granlund [20] gives a description of the curve fitting techniques used and how the

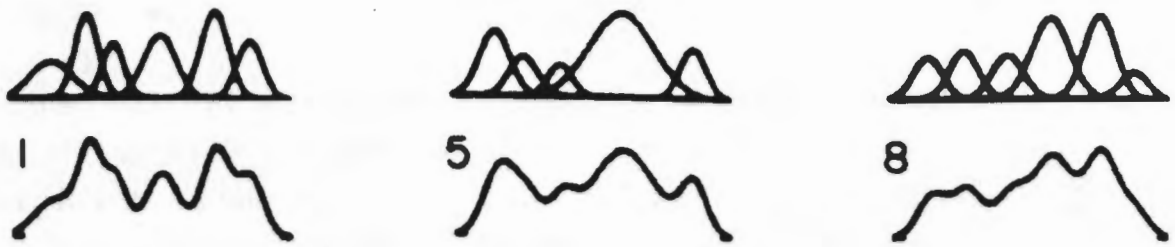


Figure 4.1: Averaged density profiles of chromosomes 1, 5 and 8 in terms of distribution functions (from [20])

Curve Matching	Sampled Curves	Fourier coefficients	Distribution functions
17.1	15.0	22.4	9.9

Table 4.2: The percentage error rates of the chromosome classification experiments performed by Granlund [20]. The results were obtained using the hold-out method.

varying number of distribution functions on different types of chromosome were taken into account by him. He used a parametric classifier to classify unknown chromosomes.

Results of Classification Experiments

The results of the classification experiments using the above techniques are presented in Table 4.2. The experiments were carried out using the hold-out method with a training set of 219 chromosomes and a test set of 192 chromosomes. Bent and overlapping chromosomes and chromosomes of classes 21, 22 and Y were excluded from the sets. The number of chromosomes per class in the karyotype was not constrained. The classifications based on Fourier coefficients and distribution functions were very time-consuming when performed on the computer hardware available in the early 1970's. Therefore a preclassification stage using curve-matching was applied and only the three most likely outcomes were passed to the classification stage.

4.3.2 Classification using band transition sequences

Classification algorithm

The construction of band transition sequences is described in Section 3.4.1. Lundsteen et al. [45] used the following classification algorithm. The feature set used to describe a chromosome consisted of 28 descriptors extracted from the BT-sequence: fourteen density descriptors X_1, X_2, ..., X_{14} (each with seven possible values) and fourteen difference descriptors Y_1, Y_2, ..., Y_{14} (each with five possible values). The probability of observing each

combination of feature values was estimated using tables of absolute frequencies based on the training set.

Three global features were used: normalised logarithmic area (Z_1), area centromeric index (Z_2) and density centromeric index (Z_3). These were modelled using a Gaussian probability distribution function for each Z, where the means μ_k and standard deviations σ_k of the features were calculated using all chromosomes of type k in the training set.

The features were all assumed to be independent and the class-conditional probability for a chromosome of type k was

$$P(X,Y,Z|k) = P(X_1,\ldots,X_{14},Y_1\ldots Y_{14},Z_1,Z_2,Z_3|k) \quad (4.2)$$

$$= \prod_{i=1}^{14}[P(X_i|k)\,P(Y_i|k)]\,[P(Z_1|k)]^3\,[P(Z_2|k)]^2\,[P(Z_3|k)]^2 \quad (4.3)$$

The weighting of the global features was chosen heuristically to obtain improved balance between local and global features. Bayes' theorem (Section 2.2) was used to estimate the posterior probability. All the prior probabilities were assumed to be equal. Unknown chromosomes were classified based on the highest posterior probability, and the SE algorithm (Section 4.1.1) was used to limit the number of chromosomes per class.

Results of classification experiments

Experiments were initially done by Lundsteen et al. [45], who applied the algorithm to the Edited Copenhagen database, and repeated by Gerdes and Lundsteen [14], who used the Edited Copenhagen and CDAR databases. Gerdes and Lundsteen used a slightly improved algorithm where the global features were not assumed to be independent.

Gerdes and Lundsteen used cross-validation with two halves of the Edited Copenhagen data set to obtain an average error rate inclusive of rejected chromosomes of 3.2%. Two experiments using cross-validation to determine the error rate were performed on the CDAR data set. For the first experiment, the data set was divided into even and odd numbered metaphases, and for the second experiment, into the first and second halves. The average classification error of both experiments was 17.8%.

4.3.3 Correlation techniques

Forabosco et al. [12] classified chromosomes by using correlation coefficients between standard chromosome profiles and the profiles of the chromosomes to be classified. The set of 24 standard chromosome profiles (one for each class) was constructed by averaging a set of 38 length-normalised chromosomes in each class. During classification, the correlation coefficients of a profile of an unknown chromosome with standard profiles were calculated for all standard profiles with lengths differing by less than 30% from that of the unknown

chromosome, and the unknown profile was assigned to the class having the highest correlation coefficient. An experiment was carried out on 20 metaphases, and resulted in an 18.2% classification error.

4.3.4 Markov network models

Granum and Thomason used automatically inferred Markov network models to classify chromosomes. This method is described in detail in [78] and the results quoted here are presented in [22]. Intensity profiles are sampled into six intensity levels using a procedure similar to the one illustrated in Figure 3.9. These sampled profiles are then described by a "difference string" consisting of a series of symbols representing the magnitude of transitions between adjacent bands. The inference step involves building a constrained first-order Markov chain for each chromosome type using the difference strings of all the patterns in the training set.

Experiments were performed using the Edited Copenhagen data set. Only chromosomes of type 1–22 were used. A set of 200 chromosomes of each type was extracted from the middle of the data set, and this set was divided into a training set and test set, each containing 100 chromosomes of each type. Application of all the networks to each unknown chromosome was not done due to the computational expense. Instead, for the first experiment, an a priori knowledge of classification into Denver groups was assumed, and for the second experiment a simple "length-test" was carried out — candidate classes for an unknown chromosome were chosen by comparing the length of the unknown chromosome to the length range for each chromosome type. The best classification errors for the first and second experiments were 6.4% and 7.3% respectively.

4.3.5 Neural networks

Jennings and Graham [32] conducted a preliminary study into using neural networks, specifically the Kohonen self-organising feature map[1] and the multi-layer perceptron (MLP) (Section 2.5). A version of the Copenhagen set with 2904 chromosomes containing no overlapped or bent chromosomes and no Y chromosomes was used in the experiments. No constraints on the number of chromosomes per class was imposed. The data set was split into two parts and cross-validation was used. The best error rate obtained using a Kohonen network was 16.7% using an input vector of 29 values sampled over the length of the integrated density profile, an 18×18 array of output nodes and six passes of the training data consisting of 40 examples of each type of chromosome. The best error rate obtained with an MLP was 6.6% using a 17-15-23 network trained using backpropagation with momentum and gradual reduction of the gain parameter. The input feature vector consisted of 15 samples of the

[1]This network is described in many texts, for example, Hertz, Krogh and Palmer [27] pages 236–246

Data set	Seven Denver classes		Ten Denver classes	
	Best network topology	% error	Best network topology	% error
Copenhagen	2-14-7	5.4	2-24-10	7.3
Edinburgh	2-14-7	10.1	2-24-10	14.3
Philadelphia	2-14-7	14.6	2-26-10	17.4

Table 4.3: Classification error rates of the Denver classification MLPs. The results for the seven class classifier are from [11] and the results for the ten class classifier are from [10].

integrated density profile, the normalised length and the area centromeric index.

Errington and Graham [10] [11] continued this work by implementing a (15+X)-100-24 MLP classifier, where X=0, 1, 2, 7 or 10. The features used as inputs were 15 samples of the integrated density profile alone (X=0), or in combination with: length or centromeric index (X=1), length and centromeric index (X=2), or a classification into one of seven or ten Denver groups done by another neural network (X=7 and X=10). The seven Denver groups are shown in Table 1.1 and the ten groups are obtained by splitting group A into three groups (A1, A2 and A3) each containing one chromosome, and splitting group E into two groups (E1 and E2) with chromosome 16 belonging to group E1 and chromosomes 17 and 18 belonging to group E2.

The neural networks were trained using a modification of the back-propagation technique which reduces the value of the learning rate η during training. While the back-propagation algorithm is running, the network classification performance on the training set and the error term over each epoch are monitored. The gain term is halved if the classification error rate does not decrease after four epochs or if the error term increases by 10% over the value at the previous epoch.

The classification into Denver groups was done using a neural network with 2 inputs (length and centromeric index) and 7 or 10 outputs. The classification errors of the Denver classification networks with the best performance are given in Table 4.3. The classification errors for the networks which classify a chromosome into one of 24 classes are given in Table 4.4. No constraints on the number of chromosomes per class were imposed.

Lerner et al. [43] used a MLP to classify cells obtained from the Institute of Medical Genetics at the Soroka Medical Centre in Beer-Sheva, Israel. The cells were divided into a "superior" set and an "inferior" set based on the chromosome quality. Only chromosomes of classes 2, 4, 13, 19 and X were used in the experiments. They compared the skeletonisation and piecewise linear (PWL) methods of extracting the chromosome axis. Features passed to the neural network were various combinations of 64 integrated density profile values extracted from the axis, length and length centromeric index. Each feature was normalised to lie in the [-0.5, 0.5] range.

The number of hidden units in the MLP and the initial weights were set according to

X	Extra Features	Percentage Classification Error		
		Copenhagen	Edinburgh	Philadelphia
0	None	8.8	22.3	28.6
1	Normalised length	8.4	19.4	27.6
1	Centromeric Index	7.7	21.0	26.5
2	Normalised length + centromeric index	6.9	18.6	24.6
7	Seven Denver groups	5.8	17.0	22.5
10	Ten Denver groups	6.2	17.8	22.7

Table 4.4: The percentage classification errors of a (15+X)-100-24 MLP, where the feature vector consisted of 15 samples of the integrated density profile and the extra features indicated. For the X=7 and X=10 cases, the output of an MLP which preclassified the chromosomes into one of seven or ten Denver classes was used. The results for the X=0, 1 and 2 cases are available in [10] and [11], the results for the X=7 case are available in [11], and the results for the X=10 case are available in [10].

Feature set	Best % error rate
Integrated Density Profile	7.1
Integrated Density Profile and CI	3.5
Integrated Density Profile, CI and Length	2.8
CI and Length	7.1

Table 4.5: The best percentage error rates obtained using four feature sets passed to a MLP classifier. These results are based on the "superior" data set from the Soroka Medical Centre in Beer-Sheva, Israel (from [43]).

principal component analysis applied to the feature vectors. The network was trained using back-propagation with momentum. 70–90% of all the vectors in a dataset were chosen as training vectors, and each experiment was repeated four times with different vectors used as training vectors. The results were averaged.

It was found that extracting the axis using the skeletonisation method gave superior classification results. Tests were performed using four sets of features. These feature sets and the best error rates obtained using the "superior" data set are summarised in Table 4.5. The "inferior" data set yielded results 4–14% lower. It should be noted that comparing these results to those reported in other work can be misleading, as only a subset of chromosome classes was used. As each class used belongs to a different Denver class, the chromosome classes most likely to be confused are eliminated from the experiment.

4.3.6 Local band description

Local band description methods attempt to represent each band in a profile separately. Groen et al. [24] used a a two-dimensional local band description method to attempt to overcome the potential problems with using integrated density profiles described in Section 3.2. Bands are extracted by first thresholding each chromosome to locate regions potentially having a band. The image is then filtered using a two-dimensional Laplace filter[2] to locate each band. Bands with a size below a heuristically set threshold are discarded, and if no bands are found, the chromosome is rejected from further analysis.

For each band found, the minimum, maximum and middle position on the main chromosome axis, the area of the band and the darkness of the band are calculated. If two bands on separate chromatids coincide then they are merged.

For each chromosome, the following features are passed to the classifier: Length, centromeric index, the location of the band with the largest area, the location of the darkest band, the location of the first band after the centromere, the location of the darkest band on the p arm and the location of the darkest band on the q arm.

The best results were obtained using a Bayes classifier. The tests were performed on a non-standard version of the Copenhagen database containing 7284 chromosomes. The method resulted in an 11.5% error rate with no rejected chromosomes. It is not stated whether a constraint on the number of chromosomes per class was implemented.

Johnston, Tang and Zimmerman [33] used local band features derived from the integrated density profile. Five features are extracted for each dark band on the chromosome. Figure 4.2 diagrams the extracted features for a band between delimiters V_1 and V_2. The features are:

band-position (X): the location in pixels of the maximum height;

band-width (W): the length in pixels of the chord (V_1,V_2);

band-height (H): the grey-level distance between the maximum height and (V_1,V_2);

band-mass (M): the summed optical density between the band profile and (V_1,V_2);

band-shape (S): $S = \frac{M}{A}$, where $A = \frac{1}{2}\left((H - c_{V_1}) + (H - c_{V_2})\right)(V_2 - V_1)$, where c_{V_1} and c_{V_2} are the density values at points V_1 and V_2 respectively.

A maximum likelihood classifier (Section 2.3) was used for classification. Only chromosomes with the same number of bands were compared with each other. As chromosomes of a certain class can have a variable number of bands visible, this involved training a number of classifiers for each class. Unknown chromosomes were classified using only classifiers trained on examples containing the same number of bands. An algorithm similar to the SE algorithm (Section 4.1.1) was used to limit the number of chromosomes assigned to a class.

The classifier was tested on the Edited Copenhagen data set. The cross-validation classification error with two halves was found to be 30.5%

[2]See one of the many standard books on image processing, such as Jain [31] page 351

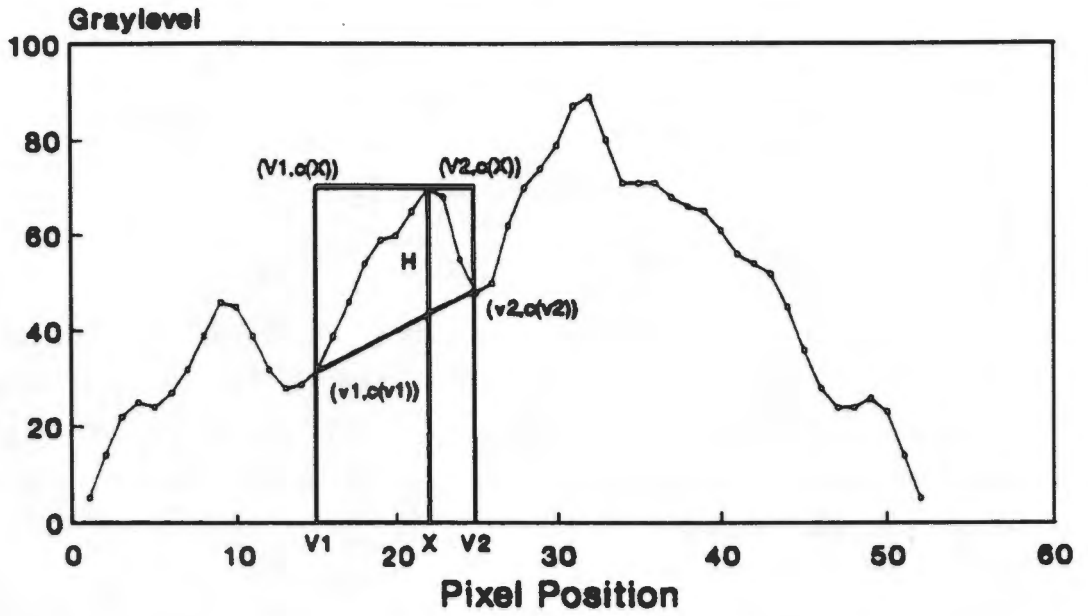


Figure 4.2: Diagram showing the five band characteristics, described in the text, used to classify chromosomes (from [33]).

4.4 Classification using weighted density distributions and other features

4.4.1 Early experiments

The first experiments using wdd's as features were carried out by Lundsteen et al. [46] and Gerdes and Lundsteen [14], who extracted eleven features from each chromosome. These were eight wdd's constructed using the weighting functions illustrated in Figure 3.11 and three global features: the normalised chromosome area, the centromeric index by area and the centromeric index by density. Classification of the chromosomes was based on parametric discriminant analysis (Section 2.3). The SE algorithm (Section 4.1.1) was used to limit the number of chromosomes assigned to a class.

This method was tested on the Edited Copenhagen and CDAR datasets. The classification error was estimated using cross-validation as described for the BTS experiments in Section 4.3.2. An average classification error of 2.0% was obtained for the Edited Copenhagen data set [14], and an average classification error of 9.0% was obtained for the CDAR data set [46].

4.4.2 Parametric classifiers

Most of the work on chromosome classification has been dedicated to applying parametric classifiers (Section 2.3) to groups of the features listed in Table 3.2, which use the wdd

Features	Database		
	Copenhagen	Edinburgh	Philadelphia
(a)	7.1	19.3	29.2
(b)	7.3	16.2	26.7
(c)	5.9	15.9	22.3
(d)	5.8	16.3	21.0

Table 4.6: The percentage classifier error rates from [65]. The following feature sets were used in each row: (a) Ten features from the pooled feature set, (b) Ten features selected from each training data set, (c) Sixteen features from the pooled feature set, (d) Sixteen features selected from each training data set. The error rates were estimated using cross-validation with the data set split into two halves.

weighting functions shown in Figure 3.12. Each of these features is assumed to have a 'true' value for each class with some Gaussian noise added [62], which justifies modelling a group of features by a multivariate Gaussian distribution. The best published results using a maximum likelihood classifier followed by a context-sensitive rearrangement procedure were obtained by Piper and Granum [65]. Groups of 10 or 16 features shown in Table 3.3 were chosen using the MSEPCOR method. The assumption of zero feature correlation was made [62], but the method used to rearrange the chromosomes in the post-processing step is not stated. The classification errors obtained using cross-validation with the data set split into two halves are shown in Table 4.6.

Much work has been carried out on making slight adjustments to the covariance matrix to speed up computation of the Mahalanobis distance and reduce over-fitting to small data sets, although the speed of modern computers obviates the need to reduce classification accuracy to gain speed. A useful result obtained by Piper [62] is that setting the off-diagonal elements of the covariance matrix to zero (i.e. assuming that all features are uncorrelated) does not reduce classification performance by very much. The advantages of this assumption in a practical system is that there are fewer parameters to estimate while training the classifier, so a smaller training set can be used.

Further algorithms aimed at reducing the time taken to calculate the Mahalanobis distance are given by Kirby et al. [36] and Kirby and Theobald [35].

4.4.3 Probabilistic neural networks

Sweeney et al. [34] used a probabilistic neural network (Section 2.6) to classify chromosomes from the Copenhagen, Edinburgh and Philadelphia data sets. Chromosomes from each of the data sets that were either touching, overlapping or unclassifiable were excluded from the experiments. All thirty features listed in Table 3.2 were passed to the PNN. The number of chromosomes in each class was taken into account by implementing an update procedure.

Database	% Error using hold-out	% Error using leave-one-out
Copenhagen	3.8	3.0
Edinburgh	16.0	15.3
Philadelphia	21.2	21.2

Table 4.7: The classification error rates of a probabilistic neural network applied to three chromosome data sets (from [34]).

Experiments were carried out using two techniques, hold-out and leave-one-out. The results are summarised in Table 4.7.

4.4.4 The transportation method

The problem of arranging the chromosomes to take the limit on the number of chromosomes per class into account was initially formulated as a linear programming problem in the form of the transportation problem by Tso and Graham [81]. Once it was shown by Kleinschmidt et al. [38] that the particular transportation problem applicable to chromosome karyotyping has an efficient solution algorithm, it was applied to karyotyping by Tso, Kleinschmidt, Mitterreiter and Graham [82]. Kleinschmidt, Mitterreiter and Piper [39] later improved on these results.

Theory

In order to optimise the cell-wide classification, the product of the likelihoods over all possible joint-allocations

$$L = \prod_{i=1}^{m} \prod_{j=1}^{n} (l_{ij})^{x_{ij}} \tag{4.4}$$

must be maximised [39] (the principle of maximum likelihood), where m is the number of classes, n is the number of chromosomes in a cell, l_{ij} is the likelihood that chromosome j belongs to class i, and x_{ij} is 1 if chromosome j is allocated to class i, and 0 otherwise.

This is equivalent to minimising $-\log L$, or

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4.5}$$

where $c_{ij} = -\log l_{ij}$. This function must be minimised under the constraints that each chromosome should be assigned to exactly one class

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \text{for } j = 1, \ldots, n \tag{4.6}$$

and that not more than two chromosomes should be assigned to each class for chromosomes of type 1 to 22

$$\sum_{j=1}^{n} x_{ij} = 2 \quad \text{for } i = 1, \ldots, 22 \tag{4.7}$$

For the X and Y chromosomes (classes 23 and 24), the constraints used in [39] are

$$\sum_{j=1}^{n} x_{ij} \leq 2 \quad \text{for } i = 23 \tag{4.8}$$

and

$$\sum_{j=1}^{n} x_{ij} \leq 1 \quad \text{for } i = 24 \tag{4.9}$$

To prevent three chromosomes being assigned to the X and Y chromosomes, a "dummy" chromosome is added to each metaphase. This chromosome has $c_{ij} = 0$ if it is assigned to class 23 or 24, and $c_{ij} = \infty$ if it is assigned to one of classes 1 to 22.

The problem is now in the form of a transportation problem, a well known problem in operations research. The transportation problem is to determine a minimum cost shipment plan which satisfies the situation described below [82]:

> Suppose we have to transport a number of units of a commodity from a given set of m sources to n destinations. We are given a_i, the number of units available at source i, and b_j, the number of units required, or the *demand* at destination j. The cost of transporting a unit quantity from source i to destination j is γ_{ij} and this cost varies linearly as the number of units transported along this route.

If each chromosome class is viewed as a source, and each chromosome as a destination having unit demand, and γ_{ij} as the negative log likelihood that chromosome j belongs to class i, then karyotyping can be solved as a transportation problem. There is an exact solution for the transportation problem with unit demand [38].

Experimental results

The best results using the transportation algorithm were obtained by Kleinschmidt, Mitterreiter and Piper [39]. In their experiments, they

1. set c_{ij} equal to the logarithm of Mahalanobis distance from an unknown chromosome feature vector to the estimated class mean vectors, instead of using estimated likelihood.
2. used sets of 16 or 30 of the features listed in Table 3.2 and the full covariance matrix.
3. used an heuristic to weight the off-diagonal elements of the covariance matrix.

Table 4.8 shows the best results obtained using the transportation method on a number of data sets. The off-diagonal elements of the covariance matrix were weighted by 0.8. The percentage misclassifications were calculated using two-part cross-validation.

Data Set	16 Features		30 Features	
	Error Rate %	Correct Cells %	Error Rate %	Correct Cells %
Copenhagen	2.0	71	2.0	69
Edinburgh	11.9	4	11.2	10
Philadelphia	14.5	13	14.7	13
600	3.6	51	2.8	60
Cpr	2.3	70	2.0	73

Table 4.8: Results of chromosome classification experiments using the transportation method applied to the logarithm of the Mahalanobis distance after weighting off-diagonal covariance matrix elements by 0.8. The values in the "Error Rate" columns are the percentage of chromosomes misclassified. The percentage of cells which had all their chromosomes classified correctly are given in the "Correct Cells" columns (from [39]).

4.4.5 Elliptically symmetric distributions

Ritter, Gallegos and Gaggermeier [69] pointed out that the joint distribution of the features of chromosomes has a tail which is not well represented by the normal distribution, and that outliers resulting from this tail are responsible for a large number of misclassified chromosomes. They assumed that the feature vectors of chromosomes were *elliptically symmetric*, so the joint distribution of the feature set of a chromosome is characterised by a mean value, a variance matrix, and a radial function, with the same radial function form used for each class. The number of chromosomes per class was taken into account using the transportation algorithm (Section 4.4.4).

Experiments were performed on the 1305 complete female cells of the Cpr data set using a normal distribution and three different forms of radial function. The errors were estimated using cross-validation, where the 1305 cells were randomly divided in ten different ways into groups of 1100 training cells and 205 test cells. The classification error varied from 2.96% when the normal distribution was used, to 1.88% using the best-performing radial function.

4.4.6 Genetic algorithms

Piper [64] formulated the chromosome classification problem as an optimization problem suitable for solution by a genetic algorithm (Section 2.7) by writing it in the form of a function H which must be minimised. He used a genetic algorithm, with cross-over and mutation rules tailored to take the nature of the karyotyping problem into account, to perform the minimization. The form of H used by Piper is

$$H = \ln\left(1 + \sum_c \left(\left(\sum_{i \in S_c} \left(\bar{l}_i - l_{ic}\right)\right) + we_c\right)\right) \tag{4.10}$$

where l_{ic} is the log likelihood that chromosome i belongs to class c, S_c represents the group of chromosomes assigned to class c, e_c is the number of extra chromosomes assigned to class c (i.e. the number of chromosomes in class c minus 2), $\bar{l}_i$ is the maximum log likelihood for a chromosome i and w is a positive constant. Minimising the first term in the summation over classes requires that chromosomes with the maximum log likelihood of belonging to class c are assigned to it. The second term in the summation over classes penalises classes which contain more than two chromosomes, where higher values of w cause the constraint to be more rigorous. Other forms of the equation expressing the same restrictions could have been chosen, but it was discovered that this form led to more rapid convergence when genetic algorithms were applied. It was also discovered that lower error rates could be obtained if some initial classifications were made by another method before the genetic algorithm was applied. The possible classes to which chromosomes could be assigned were limited to a few most likely classes in order to limit the size of the search space.

The above cost function was modified in order to take into account the similarity of the profiles of chromosomes of the same class. The revised cost function is

$$H = \ln\left(1 + \sum_c \left(\left(\sum_{i \in S_c} \left(\bar{l}_i - l_{ic}\right)\right) + we_c + p_c\right)\right) \tag{4.11}$$

where p_c is a measure of the similarity of profiles of chromosomes in class c. The similarity measures were based on cross-correlation of chromosome profiles normalised to the same length [87].

An attempt at incorporating a reject class was also made. The reject class had no class size penalty, chromosomes in the class were assumed to be perfectly matched and the log likelihood of a chromosome belonging to the reject class was taken to be its maximum log likelihood minus a system constant.

Experiments were performed on two data sets, the Copenhagen set and the 600 set, and the results obtained using two-part cross-validation are shown in Table 4.9.

Due to the computational complexity of minimising functions using genetic algorithms, this method cannot be implemented for routine use in computerised karyotyping systems in the near future.

4.4.7 Hybrid method

Kleinschmidt, Mitterreiter and Rank [40] developed a method based on a pair of classifiers. Each cell was rated as good, medium or bad based on the level of agreement between the two classifiers. The two classifiers used were the Mahalanobis-distance based method [39] described in Section 4.4.4 and a transportation method based on the l^1-norm of feature vectors weighted by variances. Applying the classification algorithm only to chromosomes in cells rated as good resulted in improved error rates which are presented in Table 4.10.

Data Set	BCF $P = 3200$	BCF $P = 800$	HM $P = 800$	rej, BCF $P = 800$		rej, HM $P = 800$	
	%E	%E	%E	&E	%R	%E	%R
Copenhagen	1.9	2.2	1.9	2.4	1.0	1.7	1.8
600	2.9	3.4	3.1	3.2	1.1	2.7	1.7

Table 4.9: Percentage classification errors obtained by applying genetic algorithms to chromosome classification [64]. P refers to the size of the population of "organisms" (strings) used by the genetic algorithm. The abbreviations used are: BCF – basic cost function (equation 4.10), HM – homologue matching (equation 4.11), rej – reject class added, %E – percentage classification error and %R – percentage of chromosomes placed in the reject class.

Data Set	% of "good" cells	% of "good" cells classified incorrectly
Copenhagen	61.1	0.58
Edinburgh	39.2	8.71
Philadelphia	36.2	7.16
600	50.7	1.65
Cpr	57.6	0.37

Table 4.10: The percentage of cells classified as good using the results of a pair of classifiers and the percentage of these "good" cells correctly classified for a number of data sets (from [40]).

This method is useful in a practical situation when an automatic metaphase finder is available. The cells found by the metaphase finder can be ranked based on the expected success of automatic classification, and only cells in the "good" group can be passed to the automatic classification stage.

4.5 Summary of the best results

Table 4.11 presents the best results obtained by methods outlined above on five data sets. Doing a fair comparison of all the results cited above is very difficult due to the slight variations in the experimental procedure followed by different research groups. It is immediately clear that the classification procedure using the transportation algorithm far outperforms the others. It is also evident that classifiers based on groups of features extracted from the chromosomes tend to outperform algorithms making direct use of sampled profiles.

Authors	Classifier	Best error rate				
		Cph	Edi	Phi	600	Cpr
Piper and Granum [65]	(a)	5.8	15.9	21.0	-	-
Kleinschmidt et al. [39]	(b)	2.0	11.2	14.7	2.8	2.0
Errington and Graham [10]	(c)	5.8	17.0	22.5	-	-
Sweeney et al. [34]	(d)	3.8	16.0	21.2	-	-
Piper [64]	(e)	1.9	-	-	2.9	-

Table 4.11: The best results obtained on the Copenhagen, Edinburgh, Philadelphia, 600 and Cpr data sets by various classifiers. The classifiers are (a) Maximum likelihood classifier using 16 features including wdd's (Section 4.4.2); (b) The transportation method (Section 4.4.4); (c) Neural Networks (Section 4.3.5); (d) Probabilistic Neural Network (Section 4.4.3) and (e) Genetic Algorithms (Section 4.4.6).

Chapter 5

Analysis of Chromosome Images Using Normalised Greyscale Correlation: Generalised Fourier Expansions of Banding Patterns

This chapter presents an analysis of one-dimensional integrated density profiles using generalised Fourier expansion methods borrowed from the theory of quantum mechanics. An advantage of using these methods is that they make use of linear mathematics only. Section 5.1 describes the construction of sets of averaged chromosome profiles. An overview of the use of generalised Fourier expansions in quantum mechanics is presented in Section 5.2, and the application of these techniques to analysing chromosome profiles and classifying chromosomes is developed in Sections 5.3 to 5.5.

5.1 Construction of library chromosomes

Libraries of 24 average chromosome profiles for the Cph and Cpr data sets were constructed in order to allow some analysis of the problem of classifying chromosomes based on profiles to be done, and for use in classification experiments. Average profiles numbered 1–22 correspond to chromosomes in classes 1–22, and the average profiles for the X and Y chromosomes are numbered 23 and 24 respectively. The libraries were constructed using the procedure outlined below. The procedure was applied to each part of each data set (a and b) separately, resulting in two sets of 24 averages being constructed for each data set.

1. An initial library was constructed using the steps below:

   (a) The orientation of reversed profiles was corrected based on the orientation flag stored with each profile in the data set.

(b) Each chromosome profile was normalised to the average length of that type of chromosome for the respective data set (shown in the tables in Appendix B) by fitting a cubic spline[1] to the profile and sampling the required number of equally-spaced points.

(c) All the length-normalised profiles of the the same class were averaged to generate 24 average profiles.

(d) The areas of each average profile were normalised.

(e) Each average profile was zero-padded with ten zeros at the beginning and zeros at the end to ensure that it was of length 128. This simplifies Fourier transforming the library chromosomes[2]. The ten zeros at the beginning were added for aesthetic reasons, as they prevent the first non-zero points of a profile from wrapping around to the end positions when a profile is shifted to the left by a small amount.

2. The initial library was processed further to improve the averages. In general, this step results in higher peaks and lower troughs.

(a) Each chromosome in the data set was correlated with the corresponding library chromosome and the position of greatest overlap between the library profile and the sampled profile was found by subpixel interpolation (this process is described in Appendix A).

(b) The position of each sampled profile in the data set was adjusted to the position of largest overlap. This was done by subtracting or adding the amount by which the profiles had to be shifted to the pixel positions (x-positions) and fitting a spline which was sampled at integer pixel positions.

(c) The shifted profiles were averaged to create new averages.

(d) The areas of each new average profile were normalised.

3. Each library profile was shifted so that their median areas were aligned, which means that

$$\sum_{i=1}^{63} P_i = \sum_{i=64}^{128} P_i$$

where P_i is the profile height at position i. The library chromosome profiles were shifted to sub-pixel accuracy by Fourier transforming them, multiplying the Fourier transform by a shifting factor and then inverse Fourier transforming them[3].

---

[1] See Press [66] Section 3.3.

[2] Algorithms which can fast Fourier transform functions with a non power-of-2 number of points do exist, but are generally not as fast.

[3] See, for example, Bracewell [5] pages 104–107 and 367

The libraries with the median areas aligned are referred to as the Cph-M and Cpr-M libraries, with the two separate sections of each library denoted by adding an "a" or "b" to the end of the name (e.g. Cph-Ma and Cph-Mb). These library averages are shown in Appendix C. On inspection of these and by comparison of the averages from the two sections of the data set, it is obvious that a number of important features are retained in the average profiles, even though Granlund [19] recommends not using averages as some information is smeared. Profile averages calculated using the separate halves of the Cph data set show a small number of differences, although there are almost no differences between the averages calculated using different halves of the Cpr data set, most probably due to the larger size of this data set.

In addition to the library described above, a second set of averaged library profiles without any length information included was constructed. This was done by carrying out steps 1 and 2 of the procedure outlined above, with the only difference being that each chromosome in the data set, regardless of class, was sampled to the average length of the class 1 chromosome for the corresponding data set. These libraries are referred to as the Cph-L and Cpr-L libraries.

5.2 A brief introduction to the use of the generalised Fourier expansion in quantum mechanics

As a number of allusions to generalised Fourier expansions in a quantum mechanical context are made in this chapter, a brief introduction to this topic is given in this section.

Quantum mechanics is a theory used to describe systems which are so small that the action of making a measurement on the system changes the state of the system. Repeated attempts to perform the same measurement on the system in order to determine the value of an observable property of the system result in different values of the measurement being returned. The state of the system therefore must be described using a statistical theory[4].

In quantum mechanics, every system is described by a state function Ψ which can be a function of space and time variables[5]. In order to simplify this discussion, consider the

[4] A more thorough treatment of the material in this section can be found in White [84].

[5] For example, a system consisting only of a single particle is described by a state function $\Psi(\mathbf{r},t)$, where $\mathbf{r}$ is the 3-dimensional position vector and t is the time. The product

$$P(\mathbf{r},t) = \Psi^{\star}(\mathbf{r},t)\,\Psi(\mathbf{r},t)$$

is interpreted as the probability that the particle will be found in the volume element between $\mathbf{r}$ and $\mathbf{r}+\mathbf{dr}$ at time t. The value of Ψ is governed by the Schrödinger equation. For example, for a particle moving in a conservative force field, the Schrödinger equation is

$$\left[-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t) = -\frac{\hbar}{i}\frac{\partial\Psi(\mathbf{r},t)}{\partial t}$$

one-dimensional time-independent system with state function $\psi(x)$. A measurement of an observable property a of the system ψ is performed by operating on it using an operator A, so $A\psi$ produces the observable a of system ψ. In order to interpret the result of the measurement, the eigenfunctions of operator A must be used. The eigenfunctions $u_n(x)$ are those functions which satisfy the relation

$$Au_n = a_n u_n \tag{5.1}$$

where a_n is the eigenvalue associated with eigenfunction u_n. As this operation is interpreted as performing a measurement on the system, the eigenvalue (the result of the measurement) must be real, which means the operator A must be hermitian. Hermitian operators have the attractive properties that their eigenvalues are real and that their eigenfunctions are orthogonal to each other, which means that the scalar product between any two eigenfunctions $\langle u_i | u_j \rangle$ is 0 if $i \neq j$. These eigenfunctions can be normalised to form an orthonormal set, which means that $\langle u_i | u_j \rangle = \delta_{ij}$, the Kronecker delta. The scalar product between two functions $p(x)$ and $q(x)$ in this one-dimensional case is defined as

$$\langle p(x) | q(x) \rangle = \int_{-\infty}^{\infty} p(x)\, q(x)\, dx \tag{5.2}$$

In order to interpret the results of applying operator A to state function ψ, an expansion of ψ in terms of the eigenfunctions of A is performed,

$$\psi(x) = \sum_n c_n u_n(x) \tag{5.3}$$

where c_n are appropriate expansion coefficients. The values of c_n are calculated using the scalar product

$$c_n = \langle u_n(x) | \psi(x) \rangle \tag{5.4}$$

which is analogous to the way in which expansion coefficients are calculated in the classical Fourier expansion in terms of $\sin nx$ and $\cos nx$, or the expansion of a function in terms of spherical harmonics.

Now consider operating on equation 5.3 with operator A. The result is

$$A\psi(x) = \sum_n a_n c_n u_n(x) \tag{5.5}$$

Further manipulation of this result shows that the expectation value of the observable $\langle a \rangle$ is

$$\langle a \rangle = \sum_n c_n^2 a_n \tag{5.6}$$

---

with Ψ normalised so that

$$\int \Psi^*(\mathbf{r},t)\, \Psi(\mathbf{r},t)\, dV = 1$$

where the integration is done over the entire region containing the particle.

This has the following interpretation: Each time the operator A is applied to system ψ to produce an observable a, one of the a_n will be produced, with each a_n being produced with a probability of c_n^2. This also implies that the system can be in state u_n with probability c_n^2. It can be shown that the values of c_n satisfy $\sum_n c_n^2 = 1$.

For example, consider a system described by the state function

$$\psi(x) = \frac{1}{2}u_1(x) + \frac{1}{2}u_2(x) + \frac{1}{\sqrt{2}}u_3(x) \tag{5.7}$$

If the operator A is applied to this system, one of three results a_1, a_2, or a_3 will be obtained. If operator A is applied to a large number of systems in this state, then a_1 will be produced $\left(\frac{1}{2}\right)^2 = \frac{1}{4}$ of the time, a_2 will be produced $\left(\frac{1}{2}\right)^2 = \frac{1}{4}$ of the time, and a_3 will be produced $\left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$ of the time. The average, or expectation, value of the observable a is

$$\langle a \rangle = \frac{1}{4}a_1 + \frac{1}{4}a_2 + \frac{1}{2}a_3 \tag{5.8}$$

5.3 Fourier analysis of chromosome profiles

In order to carry out the analysis in this section, the scalar product of two discrete profiles must be defined. Equation 5.2 defines the scalar product for continuous functions, and the discrete scalar product is defined analogously to this. The scalar product of two discretely sampled profiles p and q with lengths of 128 pixels is defined as

$$\langle p | q \rangle = \sum_{i=1}^{128} p_i q_i \tag{5.9}$$

where p_i and q_i refer to the ith pixel in the sampled profile.

Let u_i refer to the ith library chromosome profile constructed as described in Section 5.1 (where $i = 1, 2, \ldots, 24$, with chromosome X labelled 23 and chromosome Y labelled 24), and let s be the profile of a chromosome belonging to an unknown class. The profiles are normalised to have unit area using an L2-norm[6] so that $\langle u_i | u_i \rangle = 1$; $0 \leq \langle u_i | u_j \rangle \leq 1$ for $i \neq j$; and $0 \leq \langle s | u_i \rangle \leq 1$ for any chromosome profile s and all 24 library chromosome profiles. To simplify notation, define $S_{ij} = \langle u_i | u_j \rangle$ and $r_j = \langle s | u_j \rangle$.

In analogy with the representation of a functions in terms of a set of basis functions described in the previous section, one can attempt to represent s as a linear combination of library chromosome profiles

$$s = \sum_{j=1}^{24} \gamma_j u_j + \epsilon \tag{5.10}$$

where γ_j are the expansion coefficients and ϵ is the difference between the representation of s in terms of library chromosome profiles and the actual s.

[6] A profile p is L2-normalised if $\sqrt{\langle p | p \rangle} = 1$.

In order to obtain the values of the coefficients γ_j which provide the best representation of s, one minimises the mean squared error $E = \langle \epsilon | \epsilon \rangle$ with respect to the coefficients γ_j. When solving this equation to obtain values for γ_j, one would hope that if s is very similar to library chromosome i, then γ_i would be large and γ_j would be small for $j \neq i$. In the case with only two library chromosomes, minimising the error is straightforward,

$$E = \langle \epsilon | \epsilon \rangle = \int \epsilon^2 dx = \int \left(s - \sum_{j=1}^{2} \gamma_j u_j \right)^2 dx \tag{5.11}$$

$$= 1 - 2\gamma_1 r_1 - 2\gamma_2 r_2 + \gamma_1^2 + \gamma_2^2 + 2\gamma_1 \gamma_2 S_{12} \tag{5.12}$$

Taking the partial derivatives $\frac{\partial E}{\partial \gamma_1}$ and $\frac{\partial E}{\partial \gamma_2}$ and setting them equal to zero, one obtains two linear equations which can be written in the form

$$\begin{pmatrix} 1 & S_{12} \\ S_{12} & 1 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \tag{5.13}$$

One can attempt to expand this to use a larger basis containing 24 average profiles by doing a similar (but lengthier) calculation (although in practice, the use of such a large basis is not possible). If one defines a matrix S having S_{ij} in row i and column j, a vector $\boldsymbol{\gamma}$ containing γ_1 to γ_{24}, and a vector $\mathbf{r}$ containing r_1 to r_{24}, then the set of linear equations obtained when solving the 24-class problem is

$$S\boldsymbol{\gamma} = \mathbf{r} \tag{5.14}$$

It should be noted that the matrix S is real, symmetric (hermitian), diagonally dominant and positive definite. Making use of the quantum mechanical analogy again, one could interpret

$$P_i = \frac{\gamma_i^2}{\sum_{j=1}^{24} \gamma_j^2} \tag{5.15}$$

as the probability that the profile s matches library chromosome profile i. A certain amount of theoretical justification for this can be obtained by examining the case where only two library chromosomes are used. The solution to equation 5.13 is simply

$$\gamma_1 = \frac{r_1 - S_{12} r_2}{1 - (S_{12})^2} \tag{5.16}$$

$$\gamma_2 = \frac{r_2 - S_{12} r_1}{1 - (S_{12})^2} \tag{5.17}$$

The value of the denominator of equation 5.15 is

$$\gamma_1^2 + \gamma_2^2 = \frac{(r_1^2 + r_2^2)(1 - S_{12})^2 + 2S_{12}(r_1 - r_2)^2}{\left[1 - (S_{12})^2\right]^2} \tag{5.18}$$

and hence

$$P_1 = \frac{\gamma_1^2}{\gamma_1^2 + \gamma_2^2} = \frac{(r_1 - S_{12} r_2)^2}{(r_1^2 + r_2^2)(1 - S_{12})^2 + 2S_{12}(r_1 - r_2)^2} \tag{5.19}$$

If $r_1 = r_2$ then $P_1 = P_2 = \frac{1}{2}$ independent of the value of S_{12}. This is as expected, as in this case it is impossible to make an assignment, so the probability of each choice should be equal. As $\frac{r_1}{r_2}$ becomes indefinitely large,

$$P_1 = \frac{\gamma_1^2}{\gamma_1^2 + \gamma_2^2} \rightarrow \frac{1}{1 + S_{12}^2} \tag{5.20}$$

$$P_2 = \frac{\gamma_2^2}{\gamma_1^2 + \gamma_2^2} \rightarrow \frac{S_{12}^2}{1 + S_{12}^2} \tag{5.21}$$

so the chromosome with profile s will be always be assigned to class 1, as $1 > (S_{12})^2$, but the confidence with which it is assigned increases as the similarity S_{12} between the two library profiles decreases. The use of an expansion in only two library profiles does not give much more information than the values of r_1 and r_2, but using larger bases of library chromosome profiles should lead to some further insight due to the inclusion of more off-diagonal correlation coefficents in the matrix S. In practice, the full 24 class case cannot be used. This is discussed further in Section 5.5.

The main difficulty with this formulation of the problem is that because of the similarity of the chromosome profiles, the matrix S is badly conditioned and almost singular. The S matrices for the Cph-Ma, Cph-La, Cpr-Ma and Cpr-La libraries are shown in Appendix D, and the values of the largest and smallest numerically calculated eigenvalues and the ratio of the largest to smallest eigenvalues of the matrix S for the Cph-M and Cph-L library chromosomes are shown in Table 5.1. The large ratios of the largest to smallest eigenvalues demonstrate the high level of ill-conditioning of the problem. This demonstrates why classifying chromosomes using only profile information is a difficult problem.

An objection to the analogy of the analysis above to quantum mechanics is the fact that the functions u_i (the averaged chromosome profiles) used in the expansion in equation 5.10 are not orthogonal to each other. An attempt is made to correct this difficulty in the next section by constructing a set of orthogonal chromosome profiles.

5.4 Orthogonal chromosome profiles

The eigenvalues of matrix S are labelled ω_i, and are sorted so that $\omega_1 > \omega_2 > \ldots > \omega_{24}$. The eigenvectors corresponding to eigenvalues ω_i are $\mathbf{b}^i$. Let U be the matrix $(\mathbf{b}^1, \mathbf{b}^2, \cdots, \mathbf{b}^{24})$ which has the eigenvectors as columns and which diagonalises S such that

$$\left(U^{\mathrm{T}} S U\right)\left(U^{\mathrm{T}} \boldsymbol{\gamma}\right) = U^{\mathrm{T}} \mathbf{r} \tag{5.22}$$

and

$$SU = UW \tag{5.23}$$

Library	Smallest eigenvalue	Largest eigenvalue	Ratio
Cph-Ma	7.94×10^{-5}	1.97×10^{1}	2.48×10^{5}
Cph-Mb	1.16×10^{-4}	1.98×10^{1}	1.71×10^{5}
Cph-La	1.41×10^{-5}	2.21×10^{1}	1.57×10^{6}
Cph-Lb	9.45×10^{-6}	2.24×10^{1}	2.37×10^{6}

Table 5.1: The largest and smallest numerically calculated eigenvalues and the ratio of the largest to smallest eigenvalue of the matrix S for the indicated sets of library chromosome profiles.

where W is the 24×24 matrix with eigenvalues on the diagonal. As S is hermitian, the eigenvalues of S are real and the eigenvectors are orthogonal. The positive-definiteness of S ensures that the eigenvalues are positive.

Let $\mathbf{u}$ be a vector consisting of the L2-normalised library chromosomes $\{u_1, u_2, \ldots, u_{24}\}$. It is possible to define a set of orthogonal chromosome profiles $\mathbf{v} = \{v_1, v_2, \ldots, v_{24}\}$ by a linear transformation

$$\mathbf{v} = U^{\mathrm{T}}\mathbf{u} \tag{5.24}$$

The L2-norm of the ith orthogonal chromosome v_i is equal to the square root of the corresponding eigenvalue ω_i. To see this, consider calculating $\langle v_i | v_i \rangle$ for orthogonal chromosome i, taking equation 5.23 into account

$$\begin{aligned} \langle v_i | v_i \rangle &= \int \left(\sum_j U_{ij}^{\mathrm{T}} u_j \right) \left(\sum_j U_{ij}^{\mathrm{T}} u_j \right) dx \\ &= \sum_j U_{ij}^{\mathrm{T}} \langle u_j | u_j \rangle U_{ij} \\ &= \omega_i \end{aligned}$$

This allows one to construct a set of normalised orthonormal chromosome profiles so that $\langle v_i | v_j \rangle = \delta_{ij}$ by defining

$$v_j = \frac{1}{\sqrt{\omega_j}} \sum_{k=1}^{24} U_{jk}^{\mathrm{T}} u_k \tag{5.25}$$

These orthogonal chromosome profiles should consist of the important band-discriminating differences between the profiles of different classes of chromosome. Each chromosome s can now be expanded in terms of orthogonal chromosomes as

$$s = \sum_{j=1}^{24} c_j v_j + \epsilon \tag{5.26}$$

Minimising $\langle \epsilon | \epsilon \rangle$ with respect to c_j, one finds that the values of c_j are calculated in the same way as is shown in equation 5.4 in the quantum mechanical context,

$$c_j = \langle v_j | s \rangle \tag{5.27}$$

Once the values for all 24 c_j coefficients have been calculated, one can transform back to $\boldsymbol{\gamma}$ by using

$$\gamma_j = \sum_{k=1}^{24} U_{jk} \frac{c_k}{\sqrt{\omega_k}} \tag{5.28}$$

as the orthogonal chromosome profiles are defined in terms of the real chromosome profiles by an orthogonal transformation. Unfortunately, this conversion from the c_j to γ_j coefficients can be numerically unstable due to the division by the square-root of the eigenvalues, as the very small eigenvalues can add a large amount of noise to the sum. Even though the problem of the ill-conditioning of the S matrix has been removed by the use of orthogonal chromosome profiles, some numerical instability remains in this calculation.

5.4.1 Sum rules

One can prove two useful sum rules for the c_j and γ_j coefficients. These are

$$\sum_{j=1}^{24}\sum_{k=1}^{24} S_{jk}\gamma_j\gamma_k = \sum_{j=1}^{24} c_j^2 = 1 \tag{5.29}$$

and

$$\sum_{k=1}^{24} \gamma_k^2 = \sum_{j=1}^{24} \frac{c_j^2}{\omega_j} \tag{5.30}$$

The proof for equation 5.29 will be presented for the two-class case, and can easily be expanded to the 24 class case. Refer back to equation 5.11, the difference between a profile s and the profile constructed by summing multiples of the library profiles u_i. If one substitutes the values for r_1 and r_2 given by equation 5.13 which minimise equation 5.11 into equation 5.12, then one obtains

$$E = 1 - \gamma_1^2 - \gamma_2^2 - 2\gamma_1\gamma_2 S_{12} \tag{5.31}$$

At the best possible minimum, E should have a value of zero, and hence

$$\sum_{i=1}^{2}\sum_{j=1}^{2} S_{ij}\gamma_i\gamma_j = 1 \tag{5.32}$$

Proving the second part of equation 5.29 simply requires the replacement of γ_i and γ_j in equation 5.32 with c_i and c_j, and replacing S_{ij} with δ_{ij}. The second equality in the first sum rule (equation 5.29) assumes the completeness of the sets $\{u_j\}$ and $\{v_j\}$. The second sum rule (equation 5.30) is obtained directly from equation 5.28.

When expanding a profile in terms of orthogonal chromosome profiles, one would hope that $\sum c_j^2$ gets close to 1 after very few terms, as this would allow truncation of the Fourier series and would indicate that the most important components of a chromosome profile are well represented in the early terms. This is tested in Section 5.4.4.

5.4.2 Construction of orthogonal chromosome libraries

Orthogonal chromosome profiles were generated for the libraries of average profiles described in Section 5.1 using equation 5.25. The libraries of orthogonal profiles are referred to as Cph-OM, Cph-OL, Cpr-OM and Cpr-OL, with the two sections of each data set used to generate separate orthogonal profiles. The Cph-OMa, Cph-OMb, Cpr-OMa and Cpr-OMb orthogonal chromosome profiles are shown in Appendix E.

The generation of orthogonal functions from averages of "experimental" data, as is done here, is rather unusual. In most situations where real functions are expanded in terms of orthogonal functions, the orthogonal functions are determined using a theoretical description of the functions being represented. As no theoretical description of chromosome profiles exists, this approach had to be used.

To test the quality of these orthogonal functions, overlaps between orthogonal chromosome profiles constructed from alternate subsets of each data set were calculated. This is described in detail in Appendix F. It is evident from the matrices in Appendix F that the orthogonal profiles calculated using the Cpr data set have better inter-subset orthogonality properties than those calculated from the Cph data set, most likely due to the larger similarity between the libraries of averaged profiles calculated from the two sections of the Cpr data set. Examining the plots of the orthogonal chromosome profiles also demonstrates this, as orthogonal profiles calculated from different sections of the data sets start differing in a more pronounced way for lower numbered profiles in the Cph data set than in the Cpr data set. The large differences between higher numbered orthogonal profiles leads one to believe that they are made up largely of noise, with the noise being more prominent in the Cph data set due to the smaller number of examples in this data set.

Due to the large amount of noise in the higher numbered orthogonal profiles, it was decided not to use all 24 orthogonal profiles when using expansions in terms of orthogonal profiles to calculate values for the γ_j coefficients. Based on examination of plots of the Cph data set orthogonal profiles and values of overlaps between the orthogonal profiles, it was decided to use only the first 12 orthogonal profiles. This is further elaborated in Section 5.5.

5.4.3 Calculating the overlap between orthogonal chromosome and real chromosome profiles

Before overlaps between the Cph-OM or Cpr-OM library and real chromosome profiles can be calculated, the lengths of the real chromosome profiles must be normalised. This is done by rescaling the lengths of the real chromosome profiles by resampling using cubic splines so that the geometric mean of the profile lengths of all the chromosomes in a cell is equal to the geometric mean of the profile lengths in the corresponding library of averaged profiles. Each real chromosome profile is then shifted so that it is aligned with the orthogonal chromosome profiles. This is done by calculating the position of the maximum correlation coefficient

(to subpixel accuracy) with the first orthogonal chromosome profile, and then shifting the real chromosome profile to that position (Fourier transform shifting is used to shift the profile by subpixel amounts). The first orthogonal chromosome profile is ideal for the task of alignment as it is triangular in shape. Once the real chromosome profile has been shifted, the zero-shift overlaps with the rest of the orthogonal chromosome profiles are calculated.

Calculating overlaps of real chromosome profiles with the Cph-OL or Cpr-OL libraries is easier, as no alignment has to be done. Each real chromosome profile is interpolated using splines so that it has a length equal to that of the averaged chromosome of class 1 in the corresponding library, and the zero-shift overlaps with all the orthogonal chromosomes are then calculated.

5.4.4 Test of the sum rules on real chromosome profiles

Values of the coefficients c_j were calculated for all the chromosomes in the Cph data set using the Cph-OMa and Cph-OLa orthogonal chromosome libraries, with the coefficients calculated as described in the Section 5.4.3. As a test of the first sum rule (equation 5.29), the percentage of chromosomes of each class for which

$$K_T = \sum_{j=1}^{T} c_j^2 \tag{5.33}$$

exceeds 0.95 and 0.99 after T terms was calculated. Figures 5.1 and 5.3 show the percentage of chromosomes of all 24 classes for which K_T exceeds 0.95 for $T = 6, 12, 18$ and 24, where Figure 5.1 uses the Cph-OLa library, and Figure 5.3 uses the Cph-OMa library. Figures 5.2 and 5.4 show similar plots with a threshold of 0.99 using the Cph-OLa library (Figure 5.2) and the Cph-OMa library (Figure 5.4).

It is immediately clear from the plots that the orthogonal chromosomes do not provide a very good representation of the longer chromosomes. In Figure 5.1, almost all of the chromosomes except for chromosomes in class 1 have K_T larger than 0.95 after 12 terms. Even after 18 terms, not all the class 1 chromosomes have K_T above the threshold. Figure 5.2 demonstrates that a large number of extra terms are needed in order to increase K_T from 0.95 to 0.99.

Figure 5.3 demonstrates that the Cph-OMa library has an even poorer ability to represent the longer chromosomes. After summing 24 terms, a rather small number of chromosomes in classes 1, 2 and 3 have values of K_{24} larger than 0.95. This is most probably due to the median-area alignment of the library chromosome profiles from which the orthogonal chromosome profiles are calculated. As there are many more short chromosomes, many of the average profiles used to construct the orthogonal profiles have zero profile heights at sections which contain structure in the first three chromosome classes. This leads to less structure in these sections of the orthogonal profiles, so the beginning and end sections of the long chromosomes do not have a large palette to choose from. This is obvious from

the plots of the orthogonal chromosome profiles in Appendix E. Figure 5.3 also shows that most of the chromosomes in class 4 to class 24 have $K_{18} > 0.95$, and hence can be well represented by 18 terms. Figure 5.4 shows that very few chromosomes have values of K_T that reach the 0.99 threshold level when Cph-OMa orthogonal chromosomes are used.

5.5 Practical use of the calculated coefficients

The effect of using a minimum Mahalanobis distance classifier and replacing wdd profile features with c_j coefficients of the profiles to be classified, calculated using some of the 24 orthogonal chromosome profiles displayed in Appendix E, is investigated in Section 6.4.

In order to make practical use of the γ_j coefficients described above, a number of reductions in dimensionality of the problem have to be made. It was noted that as the average profiles have such a high similarity, use of equation 5.14 in the 24×24 case leads to nonsensical values for γ_j, as the best representation of an unknown profile in terms of 24 average profiles often involves adding a large negative multiple of one average profile to small positive multiples of other profiles, or other equally uninformative combinations. Attempts to use linear programming to force $\gamma_j \geq 0$ for all j proved fruitless. It was discovered that limiting the matrix size to 4×4 by choosing to use the four average chromosome profiles with the largest correlation coefficients with the profile to be classified leads to a more stable system in which one γ_j is usually close to one and the other three close to zero. A drawback of using a reduced set of averaged profiles is that one cannot interpret the results as probabilities (using equation 5.15) with much confidence.

Using the orthogonal chromosome profiles overcomes this limitation to some extent, although, as noted in Section 5.4.2, the orthogonal chromosomes with larger numbers tend to be somewhat arbitrary as they are highly contaminated by noise. To avoid using these noisy profiles, it was decided to use only 12 orthogonal chromosome profiles, which are calculated using a 12×12 matrix S and vector $\mathbf{u}$ in real time using the average profiles having the largest correlation coefficients with the profile of the chromosome to be classified. The choice of 12 orthogonal chromosome profiles was made after examination of the orthogonal chromosome profiles in Appendix E and the correlation coefficients between orthogonal chromosome profiles generated using different parts of the same data set shown in Appendix F. Reducing the number of orthogonal chromosomes has the additional advantage that the ratio of the largest to smallest eigenvalue of the 12×12 S matrix is one to two orders of magnitude smaller than the ratio for the 24×24 S matrix.

In Chapter 7, the use of the γ_j coefficients and normalised greyscale correlation in classifying chromosomes based solely on their banding patterns and some length information is investigated.

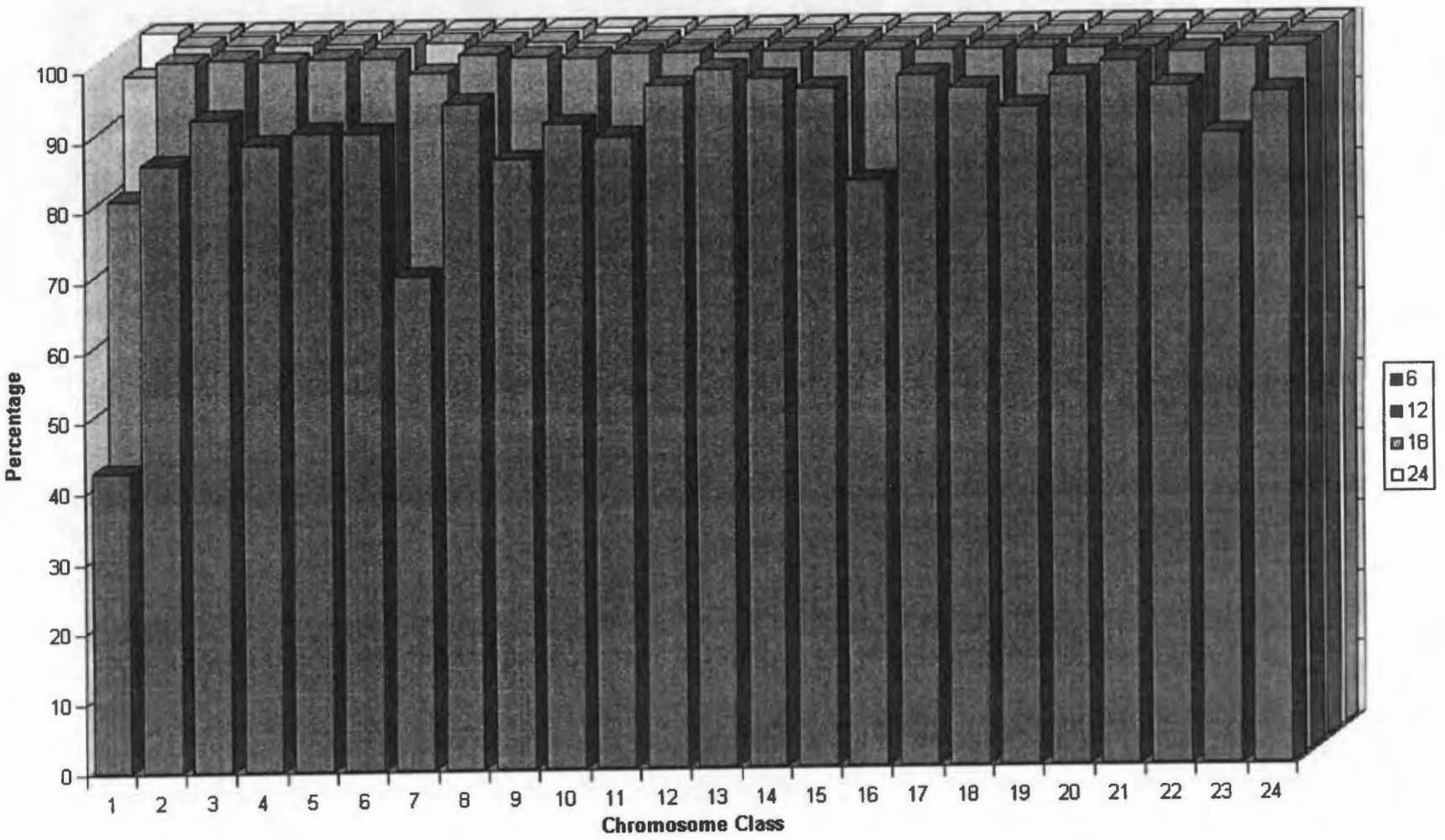


Figure 5.1: The percentage of Cph data set chromosomes of each class for which K_T (defined in equation 5.33) exceeds 0.95 after T terms when using the Cph-OLa set of orthogonal chromosome profiles, where T is shown by the colour coding indicated on the right.

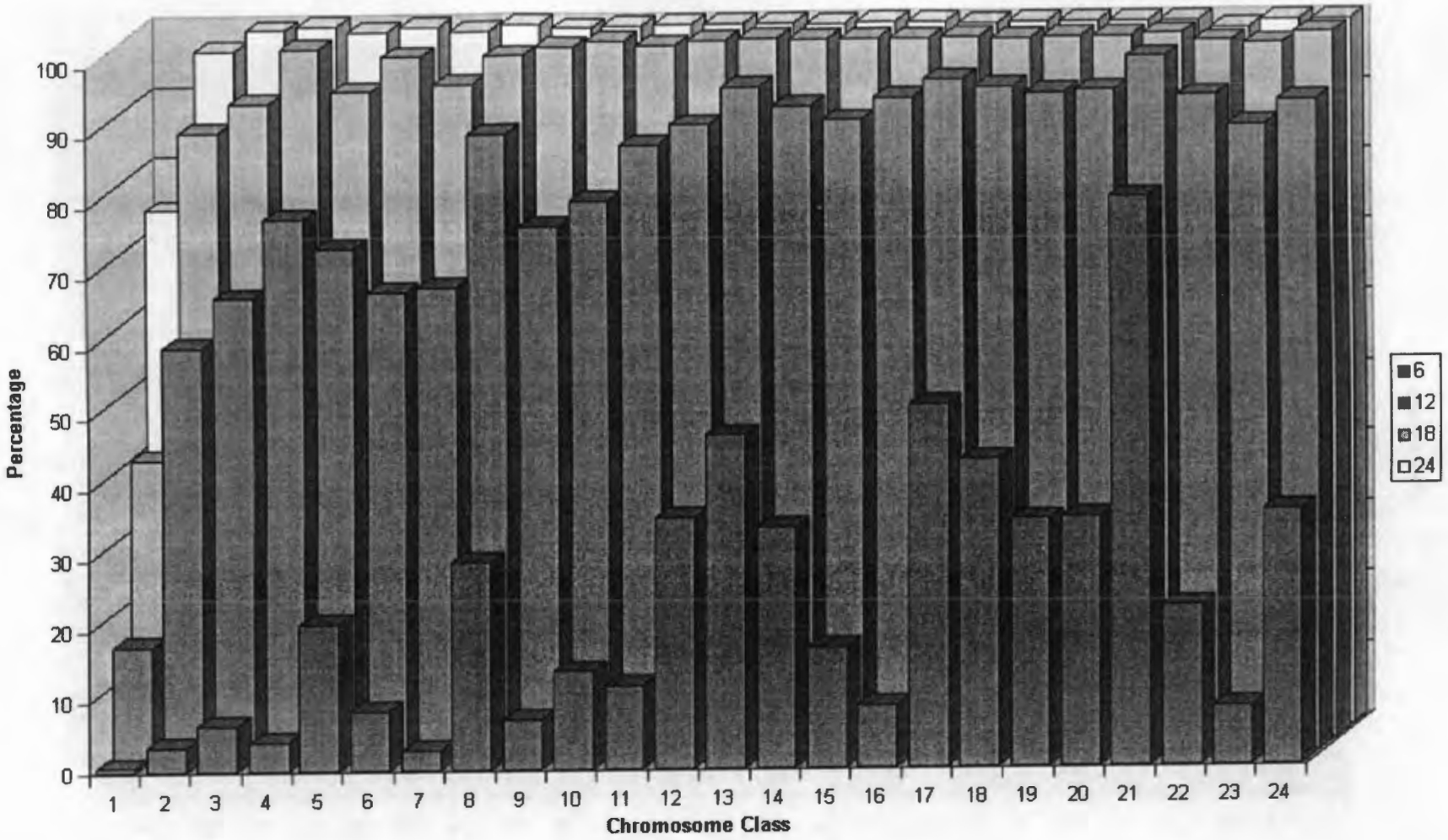


Figure 5.2: The percentage of Cph data set chromosomes of each class for which K_T (defined in equation 5.33) exceeds 0.99 after T terms when using the Cph-OLa set of orthogonal chromosome profiles, where T is shown by the colour coding indicated on the right.

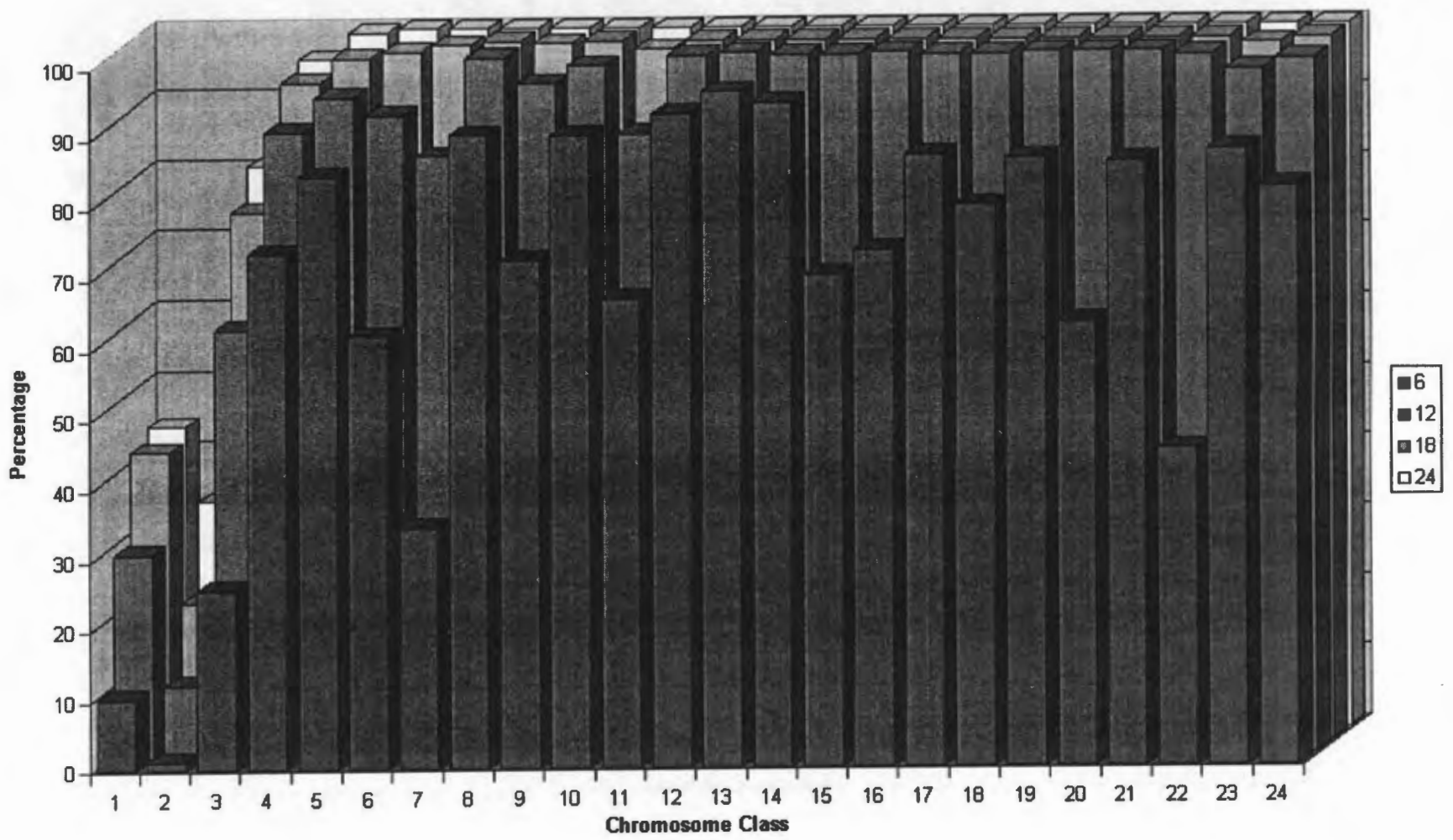


Figure 5.3: The percentage of Cph data set chromosomes of each class for which K_T (defined in equation 5.33) exceeds 0.95 after T terms when using the Cph-OMa set of orthogonal chromosome profiles, where T is shown by the colour coding indicated on the right.

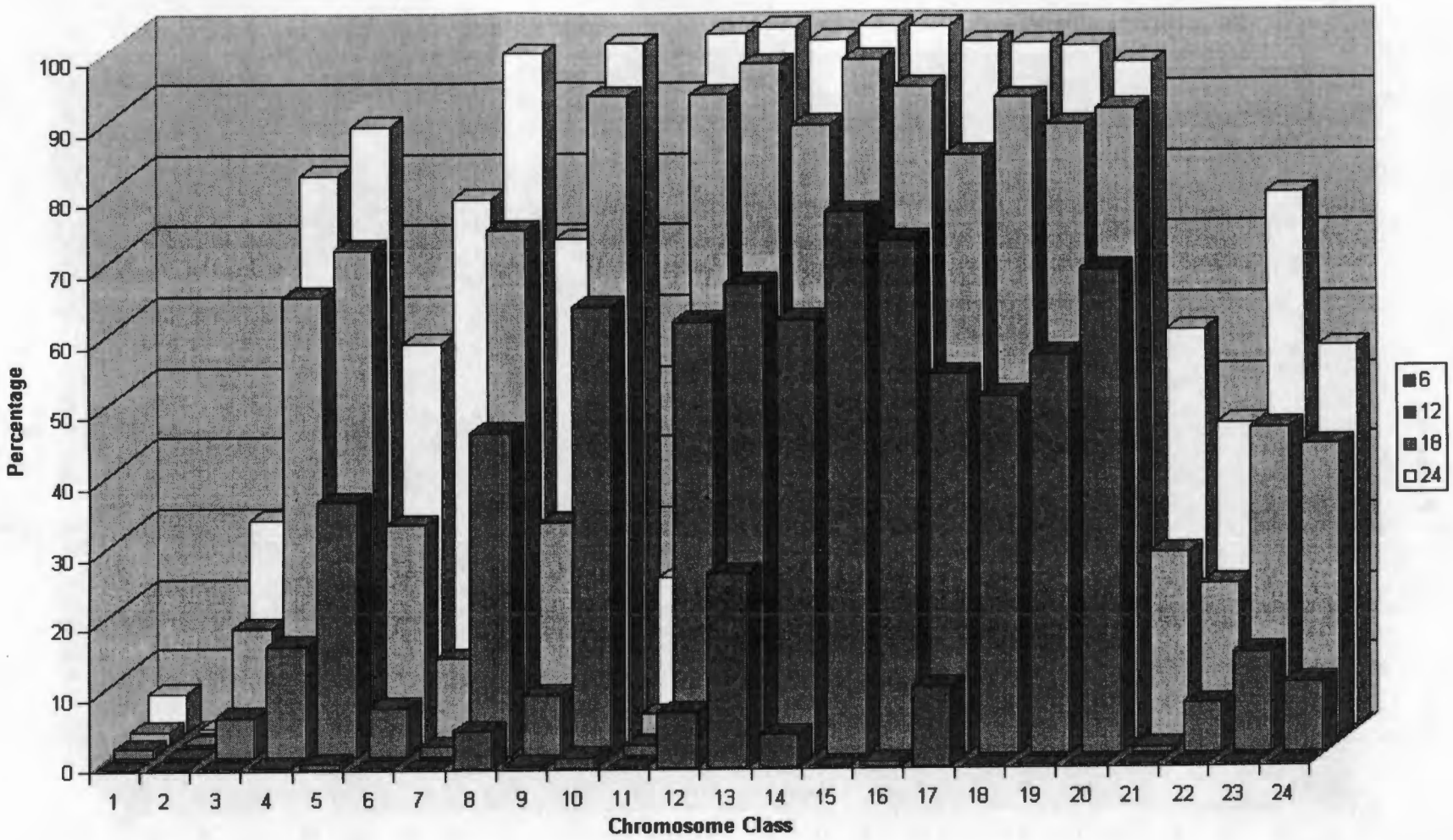


Figure 5.4: The percentage of Cph data set chromosomes of each class for which K_T (defined in equation 5.33) exceeds 0.99 after T terms when using the Cph-OMa set of orthogonal chromosome profiles, where T is shown by the colour coding indicated on the right.

Chapter 6

Experiments in Chromosome Classification: Parametric and Non-parametric Models

Experiments at classifying chromosomes into seven Denver classes, classifying chromosomes into 24 classes using the features listed in Table 3.2 and classifying chromosomes using orthogonal chromosome features are presented in this chapter. Use is made of parametric classification techniques such as the minimum Mahalanobis distance classifier, and non-parametric techniques – neural networks.

6.1 Overview of the classification techniques used

Experiments were performed on the Copenhagen (Cph), Philadelphia (Phi), Edinburgh (Edi) and Cpr data sets described in Section 1.5. Three classification techniques — k-nearest neighbour (Section 2.4.2), minimum Mahalanobis distance (Section 2.3) and neural network methods (Section 2.5) — were used in the experiments described in this chapter. Two types of neural network were used, the "cross-entropy NN" and the "sum-of-squares NN" described in Table 6.1. All neural network training was done using the simulated annealing method to set the initial weights, followed by the conjugate gradient descent method (Section 2.5.5). The simulated annealing algorithm was only used to set the initial weights, and not to escape from local minima. The parameters (defined in Section 2.5.5) used in the simulated annealing algorithm are $T_{\text{start}} = 1.0$, $T_{\text{stop}} = 0.2$, $n_t = 3$, $n_a = 50$ and $n_s = 50$. No specific stopping criterion for the training algorithm was used, and each neural network was trained for as long as possible.

The classification errors in all the experiments were calculated using two-part cross-validation (Section 2.8) in order to make the results comparable to most of the published results. The results of most classification experiments in this chapter are presented in

Network name	Error function	Hidden unit activation function	Output unit activation function
cross-entropy NN	cross-entropy	tanh	softmax
sum-of-squares NN	sum-of-squares	tanh	linear

Table 6.1: The two types of neural network used in the experiments.

\	A	B
A	[tr A, te A]	[tr A, te B]
B	[tr B, te A]	[tr B, te B]
	Average of [tr A, te B] and [tr B, te A]	

Table 6.2: The format in which results of classification experiments are tabulated. This table usually forms part of larger tables containing results of the same experiment performed on different data sets, or similar experiments with differing parameters. The notation [tr x, te y] refers to the percentage misclassification error when the classifier was trained using subset x (A or B) and tested on subset y (A or B) of the same data set.

tabular form in the format shown in Table 6.2. The percentage misclassifications of the classifier trained and tested on alternate subsets are presented with an average of these two values, as these are the standard cross-validation classifier performance measures. The percentage misclassification of the classifiers trained and tested on the same subset are included to provide an indication of the amount of over-fitting of the classifier to the training data.

When performing experiments involving the use of orthogonal chromosome coefficients as features, the results are presented in the format shown in Table 6.3 to take into account the two sets of orthogonal chromosomes calculated for each data set.

All the classification results quoted are context-independent, as no rearranging of chromosomes to take the constraint on the number of chromosomes per class into account was done. It has been shown many times that taking this constraint into account improves the classification rates, so it was felt that as the aim was to examine the performances of the classifiers themselves, implementing a rearrangement step would be an unnecessary complication.

6.2 Measurements and normalisation

The length and centromeric index measurements used in the Denver class classification experiments were obtained from the symbolic profile data files. The length of each chromosome was taken to be the number of points in the integrated density profile. This was

\	A	B	A	B
A	[or A, tr A, te A]	[or A, tr A, te B]	[or B, tr A, te A]	[or B, tr A, te B]
B	[or A, tr B, te A]	[or A, tr B, te B]	[or B, tr B, te A]	[or B, tr B, te B]
	Average of [or A, tr A, te B], [or A, tr B, te A], [or B, tr A, te B] and [or B, tr B, te A]			

Table 6.3: The format in which results of classification experiments using orthogonal chromosomes are tabulated. This table usually forms part of larger tables containing results of the same experiment performed on different data sets, or similar experiments with differing parameters. The notation [or w, tr x, te y] refers to the percentage misclassification error when the orthogonal chromosome features were calculated using orthogonal chromosomes constructed using subset w (A or B), the classifier was trained using subset x (A or B) and tested on subset y (A or B) of the same data set.

normalised within each cell by dividing each length by the median length of all the chromosomes in the cell. The machine-found centromeric indices were used in all the experiments. The normalised length and centromeric index were linearly rescaled to lie in a range approximately between -1 and 1. This was done by subtracting 1 from each normalised length, and subtracting 0.25 from each normalised centromere position and multiplying this by 4.

The 30 features in Table 3.2 were already normalised in the data sets [65]. The mean and variance of each of the 30 features were calculated for each data set using all the data in the set. Each feature was then transformed to have zero mean and unit variance by subtracting the mean from each feature variable and dividing by the standard deviation.

6.3 Software and hardware used

The k-nearest neighbour routines implemented in the SPRLIB/ANNLIB library written by the Pattern Recognition Group in the Faculty of Applied Sciences at the Delft University of Technology [29] were used. All of the neural network experiments were done using the MLFN program written by Timothy Masters [51]. A number of modifications were made to this program, including the implementation of a cross-entropy error function, softmax output activation function and a weight-decay term. Numerical routines, including the Fourier transform and correlation routines, were taken from Press et al. [66].

All experiments were run on a Pentium 133 MHz computer running the Linux operating system. All software was written in C or C++ and compiled using the Gnu C compiler version 2.7.2.1.

6.4 Comparison of classification using features and orthogonal chromosome coefficients

Due to the striking visual similarity between the wdd weighting functions and orthogonal chromosome profiles, the possibility of replacing wdd features with orthogonal chromosome coefficients was investigated. As the orthogonal chromosome "weighting functions" are derived from averages of the chromosome profiles in a data set, it is possible that they might result in better features than the wdd weighting functions. All these experiments were done using a parametric quadratic classifier, the minimum Mahalanobis distance classifier, with separate covariance matrices determined for each class. Firstly, experiments at determining the classification performance when using subsets of the features listed in Table 3.2 were performed (Section 6.4.1). A comparison of the discriminatory ability of the features extracted from the chromosome profiles (the wdd and gwdd features) and orthogonal chromosome coefficients was then carried out. The percentage misclassification when using a minimum Mahalanobis distance classifier to classify chromosomes based on subsets of the profile features in isolation was determined first (Section 6.4.2). The performance when using these subsets of profile features in combination with a selection of other features was then examined (Section 6.4.3).

6.4.1 Performance of subsets of features

A minimum Mahalanobis distance classifier was used to classify chromosomes using subsets of 10, 16, 24 and 29 of the features listed in Table 3.2. The 10 and 16 feature subsets are those chosen by the MSEPCOR algorithm applied to the pooled data sets and listed in Table 3.3 [65]. The feature vectors containing 24 features were obtained by excluding "Area", "Density c.i.", "Length", "Length c.i.", "cvhp" and "nbi" based on recommendations by Ritter et al. [69], who point out that "Area", "Length" and "cvhp" are highly correlated with "Size"; "density c.i." and "length c.i." are highly correlated with "Area c.i."; and "nbi" shows a tendency to increase error rates. The 29 feature subset was obtained by omitting "Length", the inclusion of which was discovered to lead to large increases in classifier error. The results of applying the minimum Mahalanobis distance classifier to the Cph, Edi, Phi and Cpr data sets are shown in Table 6.4.

The behaviour of the classifiers is very dependent on the size of the data set used, as expected [63]. With the three smaller data sets (Cph, Edi and Phi), some over-fitting to the training data takes place, as is evidenced by the significantly better performance of the classifiers on the training sets. As the number of features increases, the discrepancy between the performance on the training and test sets increases. With the large Cpr data set, very little over-fitting is observed. The classification error on part A of the data set is always higher than the classification error on part B, regardless of the part used to train

Number of features	\	Cph		Edi		Phi		Cpr	
		A	B	A	B	A	B	A	B
10	A	4.63	7.72	15.78	21.08	23.52	27.13	9.16	8.38
	B	6.73	5.05	21.40	14.70	30.20	21.27	9.29	8.05
		7.23		21.24		28.67		8.84	
16	A	2.58	6.18	10.16	19.00	16.05	24.07	6.66	5.84
	B	5.36	2.94	19.26	12.11	27.11	13.07	6.98	5.69
		5.77		19.13		25.59		6.41	
24	A	0.79	5.71	5.20	18.59	7.74	23.10	5.57	5.02
	B	4.16	1.28	18.49	6.41	25.08	6.43	6.11	4.80
		4.94		18.54		24.09		5.57	
29	A	0.41	6.55	3.74	19.92	6.04	24.53	5.67	5.14
	B	4.51	0.81	19.87	4.91	26.54	4.23	6.28	4.90
		5.53		19.90		25.54		5.71	

Table 6.4: The percentage of chromosomes misclassified by a minimum Mahalanobis distance classifier applied to chromosomes in the Cph, Edi, Phi and Cpr data sets using subsets of 10, 16, 24 and 29 features.

the classifier. For all four data sets, the cross-validation errors decrease as the number of features is increased from 10 to 24, and then increase when 29 features are used. The use of 29 features improves the performance of the classifier on the training sets for the three small data sets, although all four percentage misclassification rates increase for the Cpr data set.

6.4.2 Discriminatory ability of profile features used in isolation

The classification performance when classifying chromosomes based solely on features derived from the profile is examined in this section.

wdd and gwdd features

The discriminatory power of the 6 wdd and 6 gwdd features is determined by examining the classification error when applying a minimum Mahalanobis distance classifier to groups of these coefficients. The classification errors when using the indicated subsets of wdd and gwdd features are shown in Table 6.5.

Orthogonal chromosome features

Orthogonal chromosome coefficients were calculated as described in Section 5.4.3 using both the equal length (L) and median area aligned (M) sets of orthogonal profiles. No correction

Features	\	Cph		Cpr	
		A	B	A	B
wdd 1–2	A	68.44	70.94	66.97	65.96
	B	69.41	70.34	67.27	65.94
		70.18		66.62	
wdd 1–4	A	34.69	37.93	38.00	36.29
	B	35.83	35.50	38.37	36.13
		35.88		37.33	
wdd 1–6	A	18.76	21.19	21.93	19.88
	B	19.20	17.61	22.18	19.82
		20.20		21.03	
wdd 1–6 + gwdd 1–2	A	11.86	14.84	18.37	16.49
	B	13.35	11.47	18.65	16.34
		14.10		17.57	
wdd 1–6 + gwdd 1–4	A	8.64	11.66	16.05	14.31
	B	11.36	8.34	16.24	14.11
		11.51		15.28	
wdd 1–6 + gwdd 1–6	A	6.50	10.43	14.36	12.93
	B	9.66	7.16	14.72	12.61
		10.05		13.83	

Table 6.5: The percentage misclassification of chromosomes when applying a minimum Mahalanobis distance classifier to the groups of wdd and gwdd features listed in the leftmost column for the Cph and Cpr data sets.

of the orientations of unknown profiles was done (the machine determined orientation was always used). Correlations were calculated using the orthogonal chromosome profiles generated from both halves of each data set, and hence the results are presented in the format shown in Table 6.3. The classification performance when using features calculated from orthogonal chromosome profiles 1–X, where $X = 2, 4, 6, \ldots$ is presented in Table 6.6 for the Cph-OL and Cpr-OL orthogonal chromosome libraries, and in Table 6.7 for the Cph-OM and Cpr-OM libraries. The value of X was increased until adding extra orthogonal chromosome features resulted in an increase in the classification error.

Summary

The classification performance of the wdd and gwdd features and orthogonal chromosome features used in isolation is shown graphically in Figure 6.1 for the Cph data set and in Figure 6.2 for the Cpr data set. Due to the inclusion of length information in the features calculated using the M orthogonal chromosomes, these features lead to better classification performance than the other features (which do not include length information) when few features are used. Unfortunately, when more than 8 M orthogonal chromosome features are used, the classification error increases, most likely due to the uncertainty in alignment between the orthogonal chromosome profiles having large numbers of oscillations and a real chromosome profile. Although the classification performance when using the L orthogonal chromosome features is worse than that obtained using the wdd and M orthogonal chromosome features when few features are used, the performance is very similar to that obtained using 6 wdd and 6 gwdd features when more than 10 features are used. When more than 14 L orthogonal chromosomes are used on the Cph set, and more than 12 are used on the Cpr data set, the misclassification rates begin increasing.

6.4.3 Discriminatory ability of profile features used in combination with non-profile features

In this section, 18 features are passed to the classifier along with the profile features used in the previous section. The 18 features consist of those included in the 24 feature subset described in Section 6.4.1 without the six wdd features.

wdd features

The classification errors when using the 18 features in combination with 0, 2, 4 and 6 wdd features are shown in Table 6.8.

X	\	Cph				Cpr			
		Cph-OLa		Cph-OLb		Cpr-OLa		Cpr-OLb	
		A	B	A	B	A	B	A	B
2	A	75.73	75.31	75.97	77.01	75.19	73.90	74.83	73.63
	B	76.49	75.14	76.58	75.63	75.35	74.02	75.03	73.67
		76.35				74.48			
4	A	40.16	41.28	40.43	41.47	45.20	43.08	44.96	42.78
	B	41.31	38.36	41.63	38.29	45.56	43.07	45.37	42.84
		41.42				44.20			
6	A	17.59	22.03	17.54	21.83	26.06	24.02	26.02	23.94
	B	18.82	17.89	18.53	17.93	26.33	24.05	26.27	23.97
		20.30				25.14			
8	A	11.30	16.91	11.30	16.76	17.57	15.82	17.54	15.78
	B	13.38	11.75	13.00	11.64	17.83	15.71	17.76	15.67
		15.01				16.80			
10	A	7.35	11.60	7.38	11.34	14.88	13.21	14.84	13.19
	B	9.60	8.04	9.69	7.97	15.24	13.11	15.20	13.08
		10.56				14.21			
12	A	7.08	11.94	7.11	11.81	14.29	12.49	14.27	12.47
	B	9.10	7.12	8.96	6.87	14.71	12.46	14.67	12.44
		10.45				13.59			
14	A	5.71	11.02	6.00	11.15	14.59	12.78	14.56	12.75
	B	8.67	6.57	8.78	6.48	15.12	12.70	15.08	12.71
		9.91				13.93			
16	A	5.88	11.81	5.77	11.62	15.39	13.56	15.38	13.55
	B	9.34	6.59	9.16	6.61	15.95	13.42	15.93	13.39
		10.48				14.75			

Table 6.6: The percentage misclassification of chromosomes when applying a minimum Mahalanobis distance classifier to orthogonal chromosome features 1–X for the Cph and Cpr data sets. The coefficients were calculated using the Cph-OL and Cpr-OL libraries of orthogonal chromosomes.

		Cph				Cpr			
		Cph-OMa		Cph-OMb		Cpr-OMb		Cpr-OMb	
X	\	A	B	A	B	A	B	A	B
2	A	57.61	57.63	56.41	57.40	55.61	54.01	55.52	53.91
	B	57.49	56.55	57.32	56.20	55.26	53.55	55.32	53.49
		57.46				54.63			
4	A	28.66	31.07	28.13	30.17	26.99	25.39	26.90	25.36
	B	31.35	28.40	31.00	27.80	27.22	25.42	27.14	25.43
		30.90				26.28			
6	A	13.85	16.38	13.61	16.48	17.87	16.76	17.92	16.72
	B	15.84	14.33	15.90	14.26	17.93	16.20	17.95	16.22
		16.15				17.34			
8	A	8.69	14.84	8.67	16.61	14.63	14.00	14.63	14.10
	B	14.05	9.57	14.43	9.81	14.81	13.05	14.85	13.03
		14.98				14.44			
10	A	7.67	16.61	7.79	18.29	13.38	14.30	13.40	14.59
	B	14.49	9.02	15.25	9.30	15.55	11.98	15.51	11.94
		16.16				14.99			
12	A	6.62	20.60	6.70	21.02	12.39	21.40	12.41	20.22
	B	14.17	7.93	14.64	7.80	23.50	11.03	22.30	10.99
		17.61				21.86			
14	A	6.03	21.68	6.21	21.73	11.95	21.23	11.87	19.90
	B	14.34	7.68	14.75	7.85	22.88	10.41	21.99	10.36
		18.13				21.50			

Table 6.7: The percentage misclassification of chromosomes when applying a minimum Mahalanobis distance classifier to orthogonal chromosome features 1–X for the Cph and Cpr data sets. The coefficients were calculated using the Cph-OM and Cpr-OM libraries of orthogonal chromosomes.

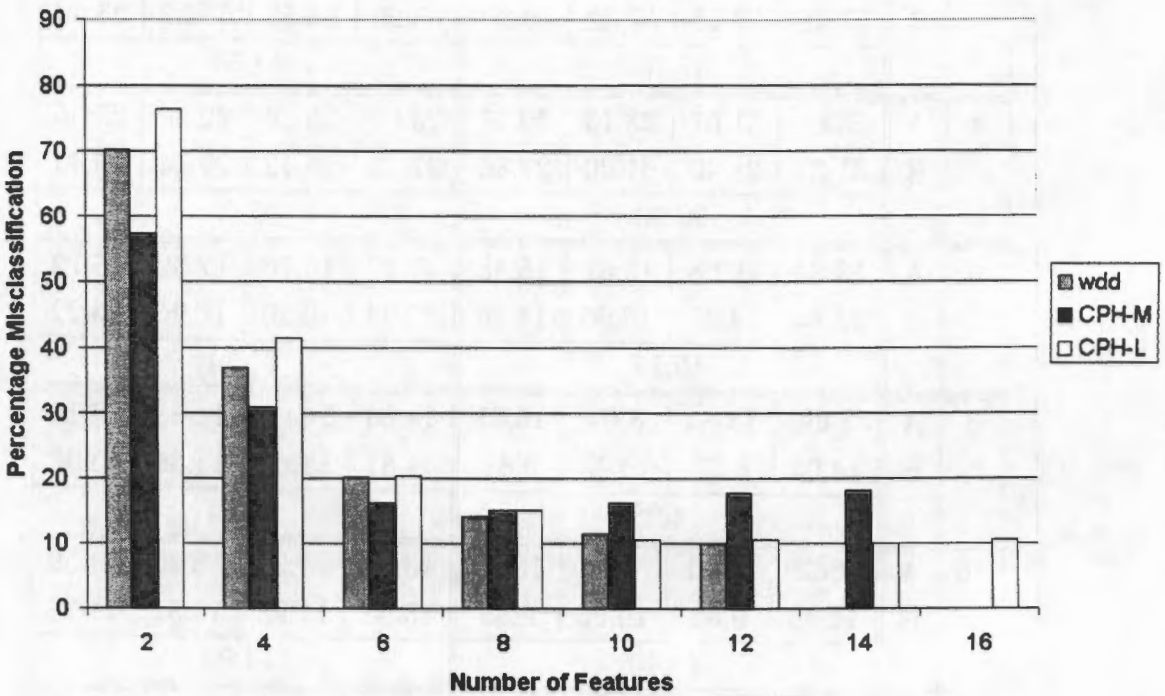


Figure 6.1: The percentage misclassification when using the indicated number of wdd or orthogonal chromosome features to classify chromosomes in the Cph data set with a minimum Mahalanobis distance classifier. The graph is constructed using the "Cph" columns in Tables 6.5, 6.6 and 6.7. The "wdd" series utilises both wdd and gwdd features as indicated in Table 6.5.

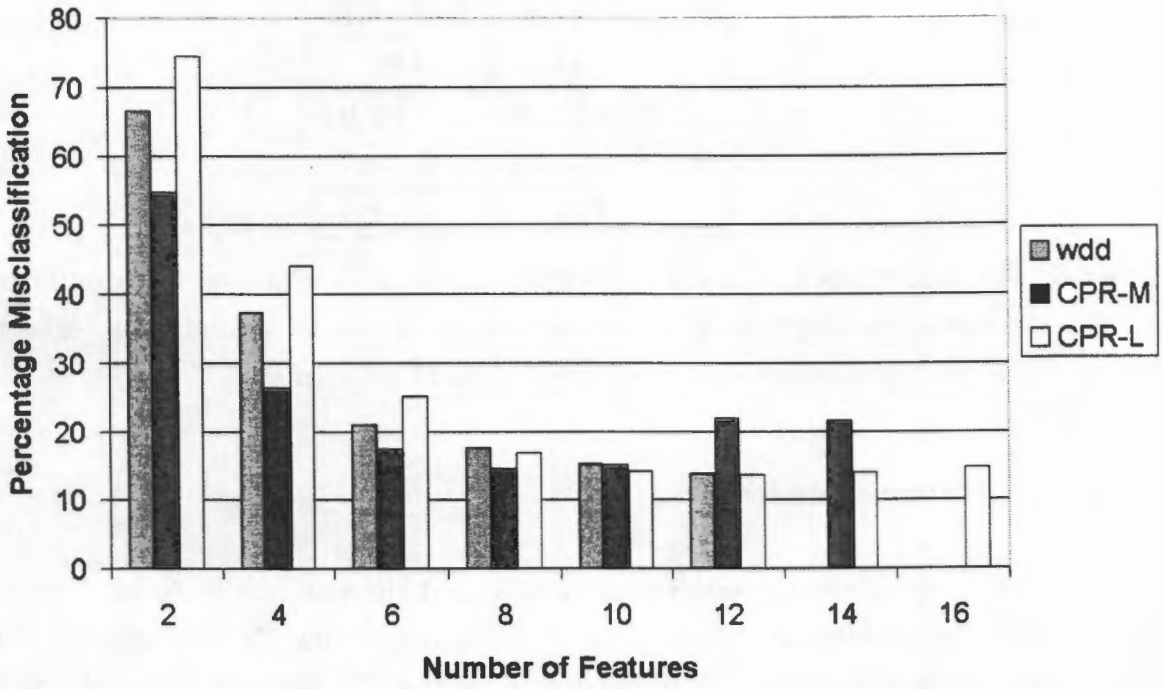


Figure 6.2: The percentage misclassification when using the indicated number of wdd or orthogonal chromosome features to classify chromosomes in the Cpr data set with a minimum Mahalanobis distance classifier. The graph is constructed using the "Cpr" columns in Tables 6.5, 6.6 and 6.7. The "wdd" series utilises both wdd and gwdd features as indicated in Table 6.5.

X	\	Cph		Cpr	
		A	B	A	B
0	A	2.37	7.80	7.07	6.34
	B	7.08	3.67	7.47	6.14
		7.44		6.91	
2	A	1.64	7.29	6.55	5.90
	B	6.32	2.45	7.01	5.71
		6.81		6.46	
4	A	1.14	6.03	5.99	5.38
	B	5.39	1.49	6.52	5.16
		5.71		5.95	
6	A	0.79	5.71	5.57	5.02
	B	4.16	1.28	6.11	4.80
		4.94		5.57	

Table 6.8: The percentage misclassification when using the 18 non-wdd features mentioned in the text in combination with X wdd features as inputs to a minimum Mahalanobis distance classifier. Results are shown for the Cph and Cpr data sets.

Orthogonal chromosome features

The classification performance when using orthogonal chromosome features 1–X in combination with the 18 features is presented in Table 6.9 for the Cph-OL and Cpr-OL orthogonal chromosome libraries, and in Table 6.10 for the Cph-OM and Cpr-OM orthogonal chromosome libraries. The value of X was increased until the addition of extra orthogonal chromosome features resulted in an increase in the classification error.

Summary

The classification performance of the wdd and gwdd features and orthogonal chromosome features used in combination with 18 other features is shown graphically in Figure 6.3 for the Cph data set and in Figure 6.4 for the Cpr data set. For the Cph data set, the use of wdd features always results in better performance than both types of orthogonal chromosome features. For both data sets, the error rate increases dramatically when more than 6 M orthogonal chromosome features are used. For the Cpr data set, the use of 10, 12 or 14 L orthogonal chromosome features leads to slightly lower misclassification rates than using 6 wdd features.

		Cph				Cpr			
		Cph-OLa		Cph-OLb		Cpr-OLa		Cpr-OLb	
X	\	A	B	A	B	A	B	A	B
2	A	1.58	7.68	1.55	7.68	6.66	5.85	6.66	5.85
	B	6.32	2.77	6.35	2.90	7.09	5.61	7.09	5.61
		7.01				6.47			
4	A	1.32	6.57	1.26	6.59	5.99	5.33	5.99	5.34
	B	4.80	1.75	4.74	1.68	6.48	5.16	6.49	5.15
		5.68				5.91			
6	A	1.05	6.38	1.02	6.31	5.73	5.09	5.72	5.06
	B	4.60	1.36	4.63	1.39	6.32	4.90	6.32	4.90
		5.48				5.70			
8	A	0.85	6.44	0.85	6.33	5.61	4.98	5.60	4.97
	B	4.33	1.13	4.39	1.17	6.22	4.80	6.20	4.79
		5.37				5.59			
10	A	0.79	6.16	0.79	6.16	5.49	4.93	5.46	4.91
	B	4.54	0.83	4.57	0.83	6.05	4.74	6.04	4.72
		5.36				5.48			
12	A	0.64	6.29	0.67	6.31	5.40	4.89	5.39	4.88
	B	4.74	0.75	4.77	0.72	6.04	4.60	6.03	4.59
		5.53				5.46			
14	A	0.47	6.18	0.47	6.25	5.49	4.85	5.49	4.85
	B	4.86	0.70	4.74	0.68	6.12	4.60	6.10	4.59
		5.51				5.48			

Table 6.9: The percentage misclassification of chromosomes when using a minimum Mahalanobis distance classifier using the 18 non-wdd features mentioned in the text in combination with orthogonal chromosome features 1–X as inputs. The experiments were performed on the Cph and Cpr data sets, and the coefficients were calculated using the Cph-OL and Cpr-OL libraries of orthogonal chromosomes.

		Cph				Cpr			
		Cph-OMa		Cph-OMb		Cpr-OMa		Cpr-OMb	
X	\	A	B	A	B	A	B	A	B
2	A	1.52	7.65	1.52	7.78	6.69	6.10	6.70	6.10
	B	6.79	3.11	6.73	3.07	7.11	5.82	7.11	5.83
		7.24				6.61			
4	A	1.32	6.87	1.38	6.95	6.19	5.64	6.20	5.66
	B	5.91	2.09	5.74	1.96	6.68	5.43	6.65	5.44
		6.37				6.16			
6	A	1.14	6.46	1.05	6.25	6.12	5.57	6.12	5.57
	B	5.91	1.86	5.77	1.81	6.52	5.29	6.49	5.31
		6.10				6.04			
8	A	1.05	8.46	0.97	10.79	6.23	5.77	6.23	5.74
	B	6.41	1.60	8.17	1.51	6.69	5.46	6.69	5.46
		8.46				6.22			
10	A	0.85	10.38	0.82	11.92	6.34	6.17	6.32	6.19
	B	6.65	1.43	9.02	1.51	7.03	5.51	7.12	5.51
		9.49				6.63			

Table 6.10: The percentage misclassification of chromosomes when using a minimum Mahalanobis distance classifier using the 18 non-wdd features mentioned in the text in combination with orthogonal chromosome features 1–X as inputs. The experiments were performed on the Cph and Cpr data sets, and the coefficients were calculated using the Cph-OM and Cpr-OM libraries of orthogonal chromosomes.

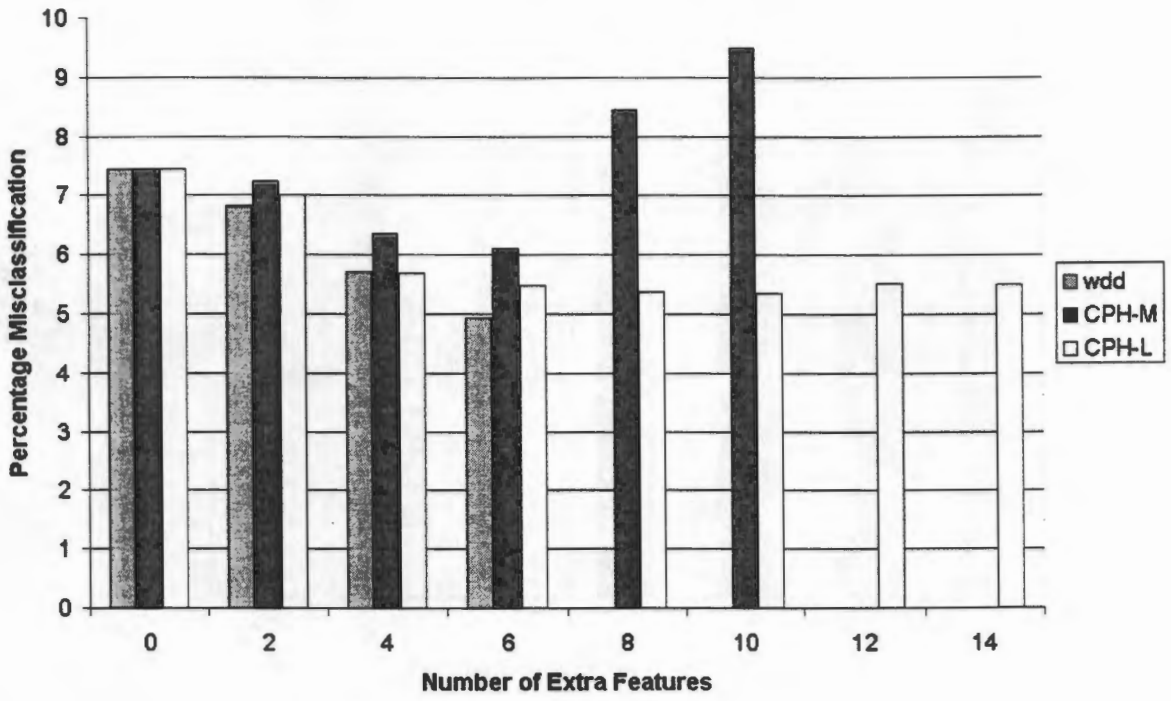


Figure 6.3: The percentage misclassification when using the indicated number of wdd or orthogonal chromosome features along with the 18 non-wdd features to classify chromosomes in the Cph data set with a minimum Mahalanobis distance classifier. The graph is constructed using the "Cph" columns in Tables 6.8, 6.9 and 6.10.

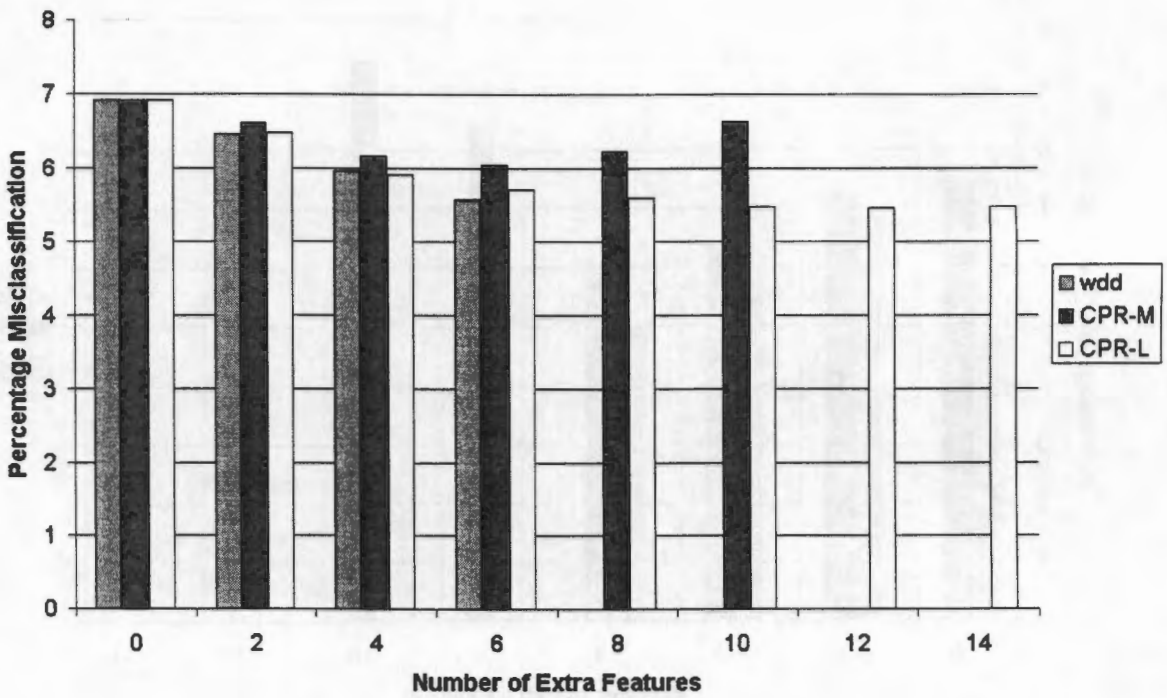


Figure 6.4: The percentage misclassification when using the indicated number of wdd or orthogonal chromosome features along with the 18 non-wdd features to classify chromosomes in the Cpr data set with a minimum Mahalanobis distance classifier. The graph is constructed using the "Cpr" columns in Tables 6.8, 6.9 and 6.10.

6.4.4 Discussion

In general, the performance of the orthogonal chromosome features used in this way was found to be disappointing. Even though the orthogonal chromosome features outperformed the wdd features when used in isolation, there is not much difference in performance when the 18 non-wdd features are included. Due to the very similar error rates when using either wdd or orthogonal chromosome features in combination with other features, there would be no particular advantage to replacing the wdd features with the correlation coefficients of orthogonal chromosome profiles with real chromosome profiles with this type of classifier.

6.5 A comparison of neural networks to traditional classification techniques

An obvious way to attempt to improve classification performance would be to introduce a non-linear classification model. The most convenient non-linear model to use is the neural network, as a large body of literature on neural networks is available. Experiments at comparing the performance of neural networks to traditional parametric classifier models were performed. The first experiment entailed the classification of chromosomes into Denver classes based on length and centromeric index information, and the second experiment attempted to classify chromosomes into twenty-four classes using the subset of 24 features found to give the best performance in Section 6.4.1.

6.5.1 Classifying chromosomes into Denver classes

Chromosomes are divided into seven Denver classes based on their length and centromeric index as described in Section 1.1. Being able to classify a chromosome into a Denver class would be a useful preclassification step before an attempt is made to classify a chromosome into one of the 24 classes. Plots of centromeric index against length (normalised as described in Section 6.2) for the Copenhagen, Philadelphia and Edinburgh data sets are shown in Appendix G. The division into seven groups is visible on the plots, although there are some characteristics which increase the difficulty of the task of classification, namely:

1. the large amount of overlap between the classes
2. the large number of outliers
3. the unreliability of the automatic centromere locating algorithm.

Four different classifiers were tried on the data sets, a k-nearest neighbour classifier, a minimum Mahalanobis distance classifier, a sum-of-squares NN and a cross-entropy NN.

The classification result of a classifier is considered correct if one of the n most probable classes with non-zero probability returned by the classifier corresponds to the correct class. Results below are given for $n = 1, 2$ and 3. This demonstrates the ability of a Denver class preclassifier to eliminate at least four Denver classes from consideration when doing a further classification into 24 classes.

The k-nearest neighbour classifier

The results obtained using the k-nearest neighbour method are shown in Table 6.11 for various values of k. Odd values of k were used to avoid ties. Testing a k-nearest neighbour classifier on the training set does not yield any useful information, and so was not done.

Mahalanobis distance classifier

The results obtained using a minimum Mahalanobis distance classifier to classify the chromosomes into Denver classes are shown in Table 6.12.

Neural networks

Pairs of 2-14-7 cross-entropy and sum-of-squares neural networks were trained on all three data sets. Initial tests, which involved repeatedly training a cross-entropy neural network starting with the same initial weights on the Copenhagen data set, were done to check the convergence and repeatability of results. It was observed that in a few tests, the neural network did not converge to the proper solution, although when it did converge, the performance was always similar. Examination of Hinton diagrams[1], which graphically depict the magnitudes of the network weights, showed that in all cases the neural network had converged to a similar state. Bad classification performance was generally caused by small anomalies in a few weights.

In the experiments reported here, the networks were trained on the data sets and then checked for convergence by testing on the training set. If they had not converged, they were retrained until convergence was achieved. Table 6.13 presents the number of epochs for which each network was trained and the classification errors.

Summary and discussion

Figure 6.5 is a graphical summary of the results obtained in this section. Unfortunately, one classifier does not stand out as being significantly better than the others, and the classification methods perform differently depending on the value of n. For the $n = 1$ case, the k-nearest neighbour classifier with $k = 15$ gives the best results for all three data sets. For the $n = 2$ case, the cross-entropy neural network performs the best on all the data sets.

[1]See Bishop [3] pages 119–120

Data Set	k	$n = 1$		$n = 2$		$n = 3$	
		tr A, te B	tr B, te A	tr A, te B	tr B, te A	tr A, te B	tr B, te A
Cph	7	5.01	5.71	2.58	2.87	2.24	2.52
		5.36		2.73		2.38	
	15	4.75	5.62	2.05	2.31	1.45	1.93
		5.19		2.18		1.69	
	31	5.18	6.24	1.75	1.99	1.11	1.32
		5.71		1.87		1.22	
	45	5.39	6.18	1.88	2.02	1.02	1.11
		5.79		1.95		1.07	
Edi	7	11.77	11.62	3.68	3.59	3.00	3.13
		11.70		3.64		3.07	
	15	10.95	11.08	2.35	2.56	1.47	1.53
		11.02		2.46		1.50	
	31	11.91	11.58	2.18	2.37	0.82	0.92
		11.75		2.28		0.87	
	45	12.42	12.07	2.25	2.33	0.68	0.65
		12.25		2.29		0.67	
Phi	7	14.07	14.91	6.53	6.69	5.00	5.03
		14.49		6.61		5.02	
	15	13.70	13.99	5.10	5.06	3.10	3.12
		13.85		5.08		3.11	
	31	13.50	14.26	4.17	4.58	2.10	2.11
		13.88		4.38		2.11	
	45	13.57	14.57	4.27	4.75	2.13	1.70
		14.07		4.51		1.92	

Table 6.11: The percentage misclassification rates for a k-nearest neighbour classifier applied to classifying chromosomes into Denver classes on the basis of normalised length and centromeric index measurements. The classifier was applied to the Cph, Edi and Phi data sets. k indicates the number of nearest neighbours used. A classification was considered correct if one of the n classes with the largest non-zero numbers of points nearest the unknown point corresponded to the correct class. Columns headed "tr x, te y" contain the classification error when the classifier was trained on part x and tested on part y of the data set. The value below each pair of values is the average (the cross-validation error).

Data Set	\	$n=1$		$n=2$		$n=3$	
		A	B	A	B	A	B
Cph	A	7.44	7.01	1.67	1.90	0.29	0.51
	B	7.67	6.97	1.84	2.11	0.56	0.51
		7.34		1.87		0.54	
Edi	A	12.46	12.32	2.10	1.88	0.27	0.31
	B	12.11	11.94	2.25	1.98	0.27	0.24
		12.22		2.07		0.29	
Phi	A	17.49	18.03	5.26	6.43	1.22	1.47
	B	16.60	15.93	5.03	5.43	1.70	1.43
		17.32		5.73		1.59	

Table 6.12: The percentage misclassification rates for a minimum Mahalanobis distance classifier applied to classifying chromosomes into Denver groups on the basis of normalised length and centromeric index measurements. The classifier was applied to the Cph, Edi and Phi data sets. A classification was considered correct if one of the n points with the smallest Mahalanobis distances from the unknown point corresponded to the correct class.

The cross-entropy neural network and the minimum Mahalanobis distance classifier produce similar results for the $n = 3$ case, with the cross-entropy neural network having the lowest classification error for the Copenhagen and Edinburgh data sets, and the Mahalanobis distance classifier having the lowest error for the Philadelphia data set. Strangely, in the $n = 1$ case for the Philadelphia data set, the minimum Mahalanobis distance classifier produced the highest error rate. As expected, the cross-entropy neural network always out-performs the sum-of-squares neural network, even though a larger number of training epochs were always used for the sum-of-squares neural network. This supports the theoretical assertion (see Section 2.5.4) that the cross-entropy neural network is better suited to classification tasks.

The high error rates for the $n = 1$ case limit the usefulness of preclassification of a chromosome into a single Denver before classifying it into one of the 24 classes. Using a Denver class preclassifier should be able to exclude at most three or four Denver classes from further consideration. Overall, the cross-entropy neural network appears to be the most successful classifier for preclassification into Denver classes. Due to the small size of the neural network, training time is not prohibitive, although the networks should always be checked for convergence after training by testing them on the training set.

Data set	Network type	No. of epochs	\	$n = 1$		$n = 2$		$n = 3$	
				A	B	A	B	A	B
Cph	cross-entropy	403	A	5.77	5.37	1.14	1.41	0.26	0.45
		275	B	5.5	4.78	1.08	1.34	0.29	0.34
				5.44		1.25		0.37	
Edi	cross-entropy	354	A	11.43	10.85	1.38	1.54	0.38	0.24
		282	B	10.62	10.20	1.49	1.33	0.31	0.17
				10.74		1.52		0.28	
Phi	cross-entropy	154	A	13.65	14.50	4.07	4.00	1.60	1.60
		125	B	14.47	13.47	4.58	4.07	1.90	1.80
				14.49		4.29		1.75	
Cph	sum-of-squares	1428	A	6.59	5.91	1.61	1.88	0.82	1.02
		1497	B	6.24	5.52	1.52	1.75	0.88	1.04
				6.08		1.70		0.95	
Edi	sum-of-squares	1298	A	12.07	11.60	2.33	2.29	0.92	0.89
		1152	B	12.04	11.53	2.22	2.12	1.49	1.13
				11.82		2.26		1.19	
Phi	sum-of-squares	11676	A	14.84	14.60	4.62	4.17	1.53	1.90
		11578	B	14.70	13.73	4.82	3.93	2.41	2.10
				14.65		4.50		2.16	

Table 6.13: The results of training cross-entropy and sum-of-squares NNs on the Cph, Edi and Phi data sets. The first three columns give information on the data set used, the network type and the number of training epochs. The final columns present the percentage classification errors of networks. The classification was considered to be correct if one of the n highest non-zero NN outputs corresponded to the correct class.

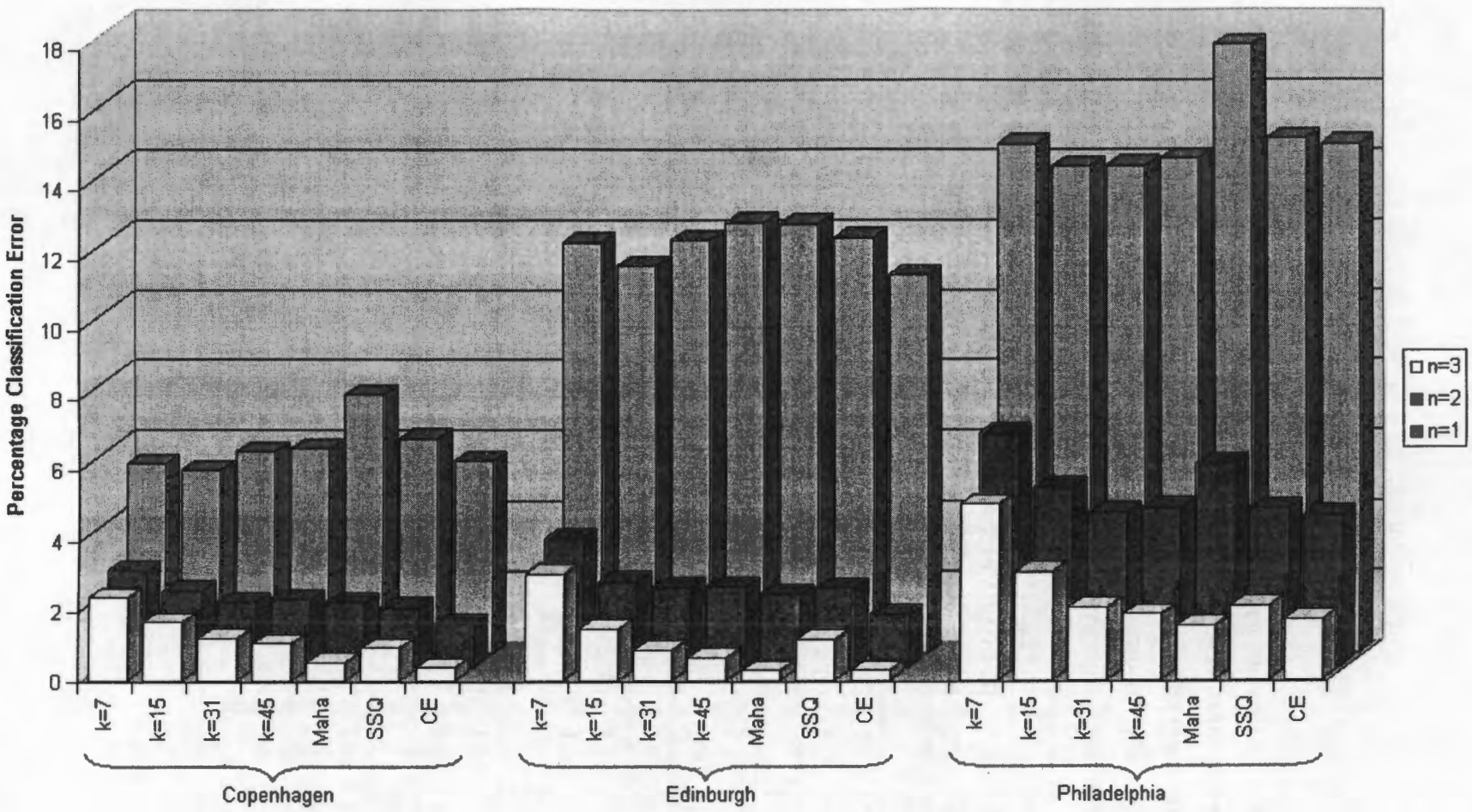


Figure 6.5: A comparison of the percentage classification errors of the various routines applied to classifying chromosomes into Denver classes based on normalised length and centromeric index measurements. The results of applying the classifiers to the Cph, Edi and Phi data sets are shown. Classifiers shown are the k-nearest neighbour classifier with $k = 7, 15, 31$ and 45; the minimum Mahalanobis distance classifier (Maha); the sum-of-squares neural network (SSQ) and the cross-entropy neural network (CE). The misclassification percentages shown are the two-part cross-validation errors from Tables 6.11, 6.12 and 6.13.

	Cph				Edi				Phi			
ν	error	\	A	B	error	\	A	B	error	\	A	B
0.0000	0.034	A	0.18	8.08	0.249	A	4.17	22.38	0.459	A	7.13	26.67
	0.059	B	8.81	0.72	0.265	B	21.36	4.57	0.345	B	28.71	4.33
			8.45				21.87				27.69	
0.0001	0.242	A	0.18	8.51	0.470	A	5.24	20.85	0.769	A	11.10	25.03
	0.290	B	7.61	1.15	0.445	B	21.55	4.67	0.721	B	27.76	8.13
			8.06				21.20				26.40	
0.0010	1.428	A	0.94	8.53	1.723	A	7.68	20.98	2.104	A	14.01	24.77
	1.410	B	9.10	2.86	1.670	B	21.63	5.97	2.027	B	28.67	11.93
			8.82				21.31				26.72	

Table 6.14: The percentage classification errors obtained by using a 24-50-24 cross-entropy neural network to classify chromosomes in the Cph, Edi and Phi data sets based on 24 features. The weight decay constant ν is given in the leftmost column. Values in the columns labelled "error" are the final training errors of the networks.

6.5.2 Classification into twenty-four classes

Due to the very good performance of the cross-entropy NN at classifying chromosomes into Denver classes, it was decided to apply it to the task of classifying the chromosomes into 24 classes using the 24 features that resulted in the best performance by the minimum Mahalanobis distance classifier. A 24-50-24 cross-entropy neural network was trained on this feature subset for the Cph, Edi and Phi data sets. Unfortunately, as shown in the top third of Table 6.14, due to the large number of parameters (weights) present in a neural network[2], use of the neural network led to even more over-fitting to the training set than was encountered with the minimum Mahalanobis distance classifier. Attempts to improve this by adding a small weight decay term (Section 2.5.6) were not very successful. A weight decay constant of $\nu = 0.0001$ reduces the classification error slightly, but the classification error increases if larger weight decay constants are used. Experiments on the Cph set using weight decay constants of 0.0100 and 0.1000 showed that this trend continues. These results are shown in the lower part of Table 6.14. Training the neural network using the Cpr data set would most likely have resulted in less overfitting, but this was not attempted due to the unreasonable amount of training time needed on such a large data set.

[2]A 24-50-24 neural network contains $[(24+1)\times 50]+[(50+1)\times 24] = 2474$ weights (the extra one added to the first term inside each set of square brackets is due to the extra bias weight). The number of weights is therefore very similar to the number of patterns in the training sets.

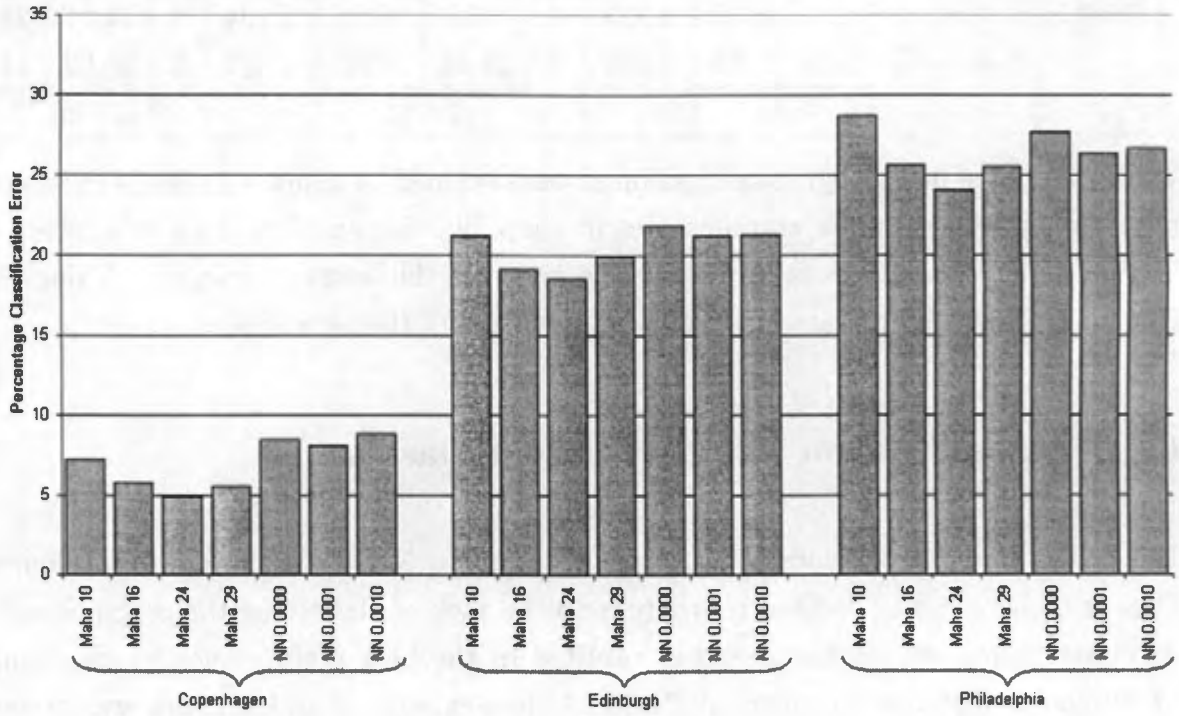


Figure 6.6: A comparison of the percentage classification errors calculated using two-part cross-validation obtained by using a number of classifiers on the Copenhagen (Cph), Edinburgh (Edi) and Philadelphia (Phi) data sets. The classifiers used are the minimum Mahalanobis distance classifier (Maha) with 10, 16, 24 and 29 features (from Table 6.4), and a cross-entropy neural network (NN) with weight decay constants ν of 0.0000, 0.0001 and 0.0010 (from Table 6.14).

6.5.3 Discussion

Figure 6.6 shows a comparison of the classification errors on the Cph, Edi and Phi data sets calculated using cross-validation for the minimum Mahalanobis distance classifier using 10, 16, 24 and 29 features and cross-entropy neural networks with weight decay constants ν of 0.0000, 0.0001 and 0.0010. It is clear that the best performance for all three data sets is achieved by using the minimum Mahalanobis distance classifier with 24 features.

Due to the disappointing performance of the neural network on the subset of 24 features, it was decided that no useful results would be obtained by attempting to use a neural network with orthogonal chromosome features as inputs. Attempts to use sampled profiles, lengths and centromeric indices as features as done in the experiments reported in Section 4.3.5 provided similar results to those presented above. The performance of neural networks cited in the literature [10] [11] is better, but ironically, use was made of a less efficient training algorithm (a modified back-propagation algorithm) along with an early stopping criterion. Due to the inherent ambiguities in deciding when to stop training, the use of early stopping was not attempted in these experiments. The use of a validation set to decide when to stop training was considered unsuitable, as this results in a change in the examples in the training set, which makes the comparison of the classification rates of the neural network with those of other classifiers less statistically sound. The only regularisation that was tried was the use of a weight decay coefficient.

In conclusion, it was decided that as neural networks are overly flexible and do not perform well when a large neural network is used with a relatively small training set, and due to the exorbitant amount time needed to train networks, it is best to use parametric models when implementing a chromosome classification routine.

Chapter 7

Classification of Chromosomes by Generalised Fourier Analysis

In this chapter, direct use is made of techniques introduced and described in Chapter 5 to classify chromosomes based on one-dimensional integrated density profiles and some length information only. It should be noted that no centromeric information is taken into account in the following experiments. A detailed description of how each of the coefficients is extracted from the profiles is given in Section 7.1 and experimental results of classification experiments using the techniques are presented in Section 7.2. A discussion of these results follows.

7.1 Description of coefficients calculated from chromosome profiles

In order to make a decision about how to classify an unknown chromosome, four sets of coefficients were calculated from the profile of the chromosome. The simplest of these are the correlation coefficients (calculated as described in Appendix A) between library chromosome profiles and the profile of the unknown chromosome (this is similar to the technique used by Forabosco et al. [12], described in Section 4.3.3). Two sets of correlation coefficients were calculated, one set which did not take the lengths of chromosomes into account at all, and one set which took a small amount of length information into account. The γ coefficients introduced in Chapter 5 were calculated in two ways, directly from an expansion of the unknown chromosome profile in terms of four library chromosomes, and by calculating an expansion of the unknown chromosome profile in terms of twelve orthogonal chromosome profiles, calculating the c coefficients, and then transforming back to γ coefficients.

A more detailed description of the calculation of each of these coefficients is presented below. The libraries of average profiles with lengths equal to the average lengths of the corresponding data set, and with median areas lined up were used (the Cph-M and Cpr-M

libraries shown in Appendix C).

1. **Correlation coefficients:** Twenty-four correlation coefficients, which do not take any length information into account, are calculated for the unknown profile. The following steps are carried out for each average profile i ($i = 1, 2, \ldots, 24$) in the library:

   (a) The length of the unknown chromosome profile is normalised to be the same as that of average profile i by fitting a cubic spline and sampling the correct number of equally-spaced points (the lengths of the average profiles for each class are shown in Appendix B).

   (b) The maximum correlation coefficient between the length-normalised unknown chromosome profile and the library average profile is determined using Fourier transform correlation (Appendix A).

2. **Length-constrained correlation coefficients**: These are calculated in the same manner as the correlation coefficients described above, except that some simple length discrimination is included. For every average profile of class i in the library, the following steps are performed:

   (a) The length of the unnormalised chromosome profile is compared with the lengths of longest and shortest chromosomes of class i in the data set (shown in Appendix B).

   (b) If the length of the unknown profile falls into this range, the maximum-overlap correlation coefficient is calculated (as above), else the correlation coefficient for class i is set to zero.

3. **Direct γ coefficients:** These are the coefficients introduced in Section 5.3 which correspond to directly setting up a representation of the unknown chromosome profile in terms of library profiles. Due to the difficulties encountered when using all twenty-four library chromosomes (mentioned in Section 5.5), they are calculated using only four library chromosomes as a basis for the expansion. The four values of γ are calculated as follows:

   (a) A 4×4 matrix of library profile overlaps (S matrix) using the four library profiles corresponding to the largest length-constrained correlation coefficients is constructed.

   (b) Values of r_j, the overlaps of the unknown profile and the four library profiles chosen based on the above criterion are calculated by:

i. doing a length normalisation on the cell to be classified, where the lengths of all the chromosome profiles in the cell are scaled so that their geometric mean is equal to the geometric mean of the lengths of the library chromosomes.

ii. shifting the unknown chromosome profile so that the position of the median area is at position 64 of the profile.

iii. setting r_j equal to the zero-shift correlation coefficient with the jth library profile.

(c) Equation 5.14 is used to calculate the four γ_j coefficients. The γ_j coefficients corresponding to classes of chromosomes which were not selected in step (a) are set to zero.

4. **γ coefficients calculated using orthogonal chromosomes:** These coefficients are introduced in Section 5.4. As mentioned in Section 5.5, making use of 24 orthogonal chromosome profiles is not advisable due to the large amount of noise contaminating the later profiles. In order to avoid this difficulty, a set of 12 orthogonal chromosome profiles is generated in real time before calculating the values of γ for the unknown chromosome. The twelve γ coefficients calculated using orthogonal chromosome profiles are determined as follows:

   (a) A 12×12 S matrix is constructed using the 12 library chromosome profiles with the highest length-constrained correlation coefficients. If less than 12 library profiles have non-zero length-constrained correlation coefficients, then other library profiles of similar lengths are used in addition to those that satisfy the correlation criterion.

   (b) Twelve orthogonal chromosome profiles are constructed using equation 5.24.

   (c) Overlaps c_k are calculated for each orthogonal chromsome by calculating the zero-shift correlation coefficient of orthogonal chromosome k with the unknown chromosome profile. Instead of using the first orthogonal chromosome to line up the unknown chromosome profile as is described in Section 5.4.3, the unknown chromosome profile with median area at position 64 is used, as this had to be determined in order to calculate the direct γ coefficients.

   (d) Equation 5.28 (with the 24 replaced by 12) is used to calculated twelve γ_k coefficients.

Even though there are two sets of γ coefficients calculated, the methods used and the different sizes of the bases result in different values of these coefficients. Different interpretations should also be attached to these two sets of γ coefficients. For the direct γ coefficients, one expects the value of one γ_j coefficient to be close to one and the other three to be close to zero, and hence converting them to probabilities does not add much insight. Converting the

γ_k coefficients calculated using orthogonal chromosomes into probabilities is more sensible due to the larger basis used.

7.2 Results of classification experiments

The first experiments involved examining the misclassification rate when classifying chromosomes based on the values of a single type of coefficient. Four initial experiments were done on the Cph and Cpr data sets, each one performing the classification based on a different type of coefficient, namely correlation coefficients, length-constrained correlation coefficients, directly calculated γ coefficients or γ coefficients calculated using orthogonal chromosomes. These coefficients were calculated using the automatically determined orientation of the chromosome profiles, so a small percentage of the profiles were inverted. In each experiment, an unknown chromosome was assigned to the class having the highest coefficient. The percentage misclassifications obtained for these experiments are shown in Table 7.1.

Further experiments involved testing the performance of combinations of these coefficients. Unknown chromosomes were assigned to a class only if the highest values of various groups of coefficients corresponded to the same class, else the chromosome was labelled as unclassified (placed in a reject class). The concept of a reject class has been used before, but in a slightly different way, linked to a constraint on the number of chromosomes placed in each class rather than on a number of different classifiers agreeing on the same class. This form of reject class is used in the SE algorithm [46] described in Section 4.1.1, and in the genetic algorithms [64] described in Section 4.4.6. Another multi-classifier approach was taken by Kleinschmidt et al. [40], who used the level of agreement between two classifiers in order to rank the quality of whole cells, rather than for placing individual chromosomes in reject classes. This method is described in Section 4.4.7.

Table 7.2 shows the percentage of the chromosomes that were misclassified and left unclassified when using the indicated combinations of coefficients. The averages of the percentages of chromosomes misclassified and unclassified for the four combinations of coefficients used are depicted graphically in Figure 7.1 for the Cph data set, and in Figure 7.2 for the Cpr data set.

7.3 Discussion

Classifying a chromosome using the extremely simple approach of assigning it to the class for which the correlation coefficient between the chromosome profile and the average profile for that class is highest leads to a surprisingly low percentage of misclassified chromosomes, as can be seen in the top row of Table 7.1. This is remarkable as no other information

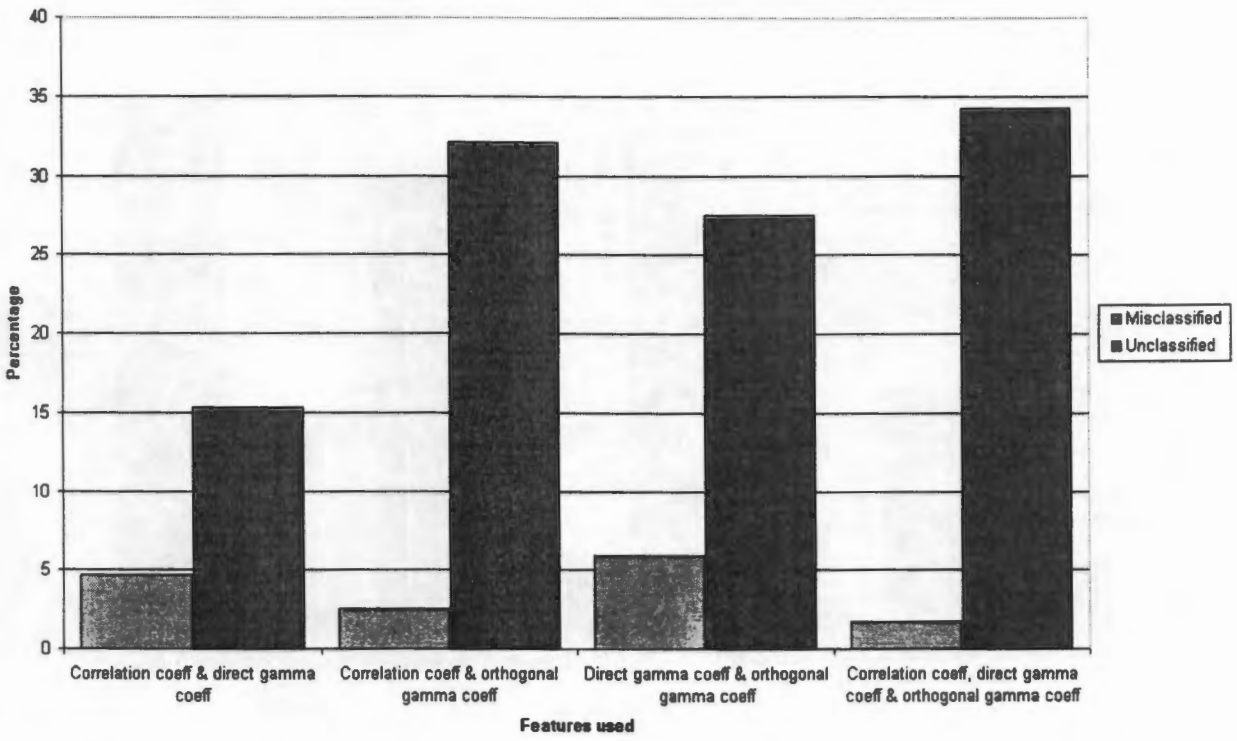


Figure 7.1: The percentage of Cph data set chromosomes misclassified and unclassified when assigning chromosomes to a class only when there is agreement between the classes suggested by the coefficients indicated below each pair of bars, and leaving the chromosome unclassified if the coefficients suggest different classes. This is a graphical representation of data in Table 7.2.

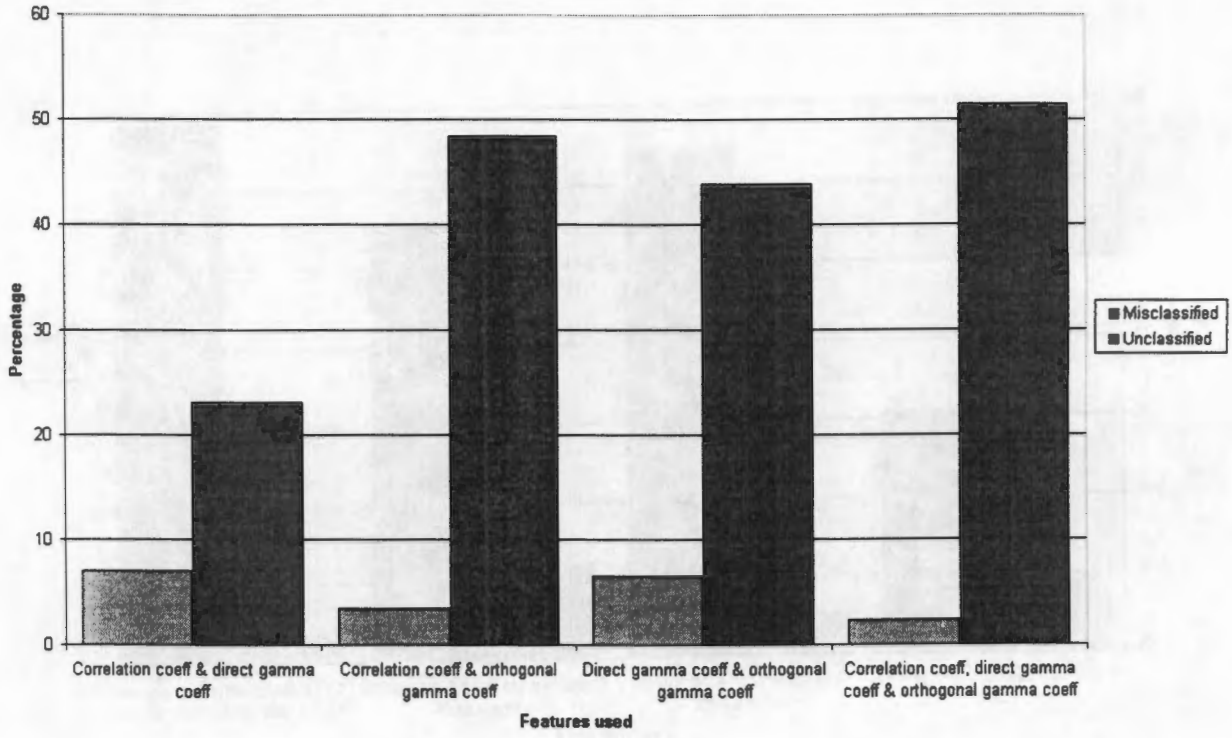


Figure 7.2: The percentage of Cpr data set chromosomes misclassified and unclassified when assigning chromosomes to a class only when there is agreement between the classes suggested by the coefficients indicated below each pair of bars, and leaving the chromosome unclassified if the coefficients suggest different classes. This is a graphical representation of data in Table 7.2.

Coefficient	\	Cph		Cpr	
		A	B	A	B
Correlation coefficients	A	13.88	15.91	24.93	22.86
	B	14.87	14.24	24.71	22.47
		15.39		23.79	
Length-constrained correlation coefficients	A	11.59	12.94	21.40	19.36
	B	12.06	11.39	21.38	19.20
		12.50		20.37	
Direct γ coefficients	A	15.60	15.61	23.78	22.41
	B	15.98	14.41	23.77	22.09
		15.80		23.09	
γ coefficients calculated using orthogonal chromosomes	A	30.47	29.30	47.55	46.21
	B	31.65	28.85	47.24	45.61
		30.48		46.73	

Table 7.1: The percentage misclassifications when classifying chromosomes in the Cph and Cpr data sets by assigning them to the class with the highest value of the coefficient indicated in the leftmost column. The results are presented in the format shown in Table 6.2.

apart from the chromosome profile is taken into account, even information about the chromosome length is discarded. Including some length information by using length-constrained correlation coefficients reduces the error by around 3% for both data sets (second row of Table 7.1). At first glance, it does not appear as if the direct γ coefficients provide much extra information, as the misclassification rates are similar to those obtained using the correlation coefficients only (third row of Table 7.1). Classifying chromosomes based on the highest γ coefficient calculated from orthogonal chromosomes is very unreliable, as the misclassification rates are twice as large as those for correlation coefficients (fourth row of Table 7.1).

However, as can be seen from the results presented in Table 7.2, making a classification decision based on a combination of these coefficients leads to a very small number of misclassified chromosomes. In the case where a chromosome is only assigned to a class if the highest length-constrained correlation coefficient, direct γ coefficient and γ coefficient calculated using orthogonal chromosomes correspond to the same class, the cross-validation misclassification rate is only 1.75% for the Cph data set, and 2.36% for the Cpr data set. This low misclassification rate is coupled with a large percentage of chromosomes left unclassified, 34.24% for the Cph data set and 51.44% for the Cpr data set, but this demonstrates that the routine can make informed decisions on whether a chromosome can be classified based purely on profile information or if other information is required. If one considers that it is generally preferred, especially in medical applications, that automated classifiers

Coefficients Used	\	Cph				Cpr			
		A		B		A		B	
LC & DG	A	4.89	14.40	4.61	15.52	7.21	24.07	6.71	22.28
	B	4.83	15.16	4.16	14.07	7.24	23.99	6.59	22.10
		4.72		15.34		6.98		23.14	
LC & OG	A	2.81	31.50	2.54	31.15	3.64	49.46	3.32	47.77
	B	2.49	33.05	2.20	30.41	3.69	49.10	3.30	47.28
		2.52		32.10		3.51		48.44	
DG & OG	A	5.83	27.72	5.35	27.10	6.84	44.49	6.38	43.38
	B	6.44	27.96	5.05	26.25	6.82	44.35	6.37	42.82
		5.90		27.53		6.60		43.87	
LC, DG & OG	A	1.99	33.67	1.70	34.95	2.46	52.39	2.27	50.61
	B	1.79	33.52	1.54	32.37	2.45	52.26	2.24	50.17
		1.75		34.24		2.36		51.44	

Table 7.2: The percentage of chromosomes in the Cph and Cpr data sets misclassified (left of each table cell) and unclassified (right of each table cell) when assigning chromosomes to a class only when there is agreement between the classes suggested by the coefficients indicated in the leftmost column, and leaving the chromosome unclassified if the coefficients suggest different classes. The following abbreviations are used for the types of coefficients: LC – length-constrained correlation coefficients; DG – direct γ coefficients; OG – γ coefficients calculated using orthogonal chromosomes. The format of the results is similar to that shown in Table 6.2, except that averages of both the percentages of misclassified and unclassified chromosomes are shown below each quartet of results.

not make rash decisions based on too little data, then this large percentage of unclassified chromosomes is tolerable (but slightly disappointing). It is interesting to note that the percentage of unclassified chromosomes is larger for the Cpr than for the Cph data set, reflecting the lower quality of the chromosomes in the Cpr data set.

When examining the performance of pairs of coefficients, one sees that the combination of length-constrained correlation coefficients and γ coefficients calculated using orthogonal chromosomes leads to results similar to using all three coefficients, which supports the assertion that most of the useful classification information is due to the use of orthogonal profiles. The use of length-constrained correlation coefficients and direct γ coefficients in combination leads to the smallest percentage of chromosomes left unclassified, but a relatively large misclassification rate. Using a combination of the two types of γ coefficient leads to the worst results, with the highest misclassification rate, and a very large percentage of chromosomes left unclassified.

A further advantage of the γ coefficients is the possibility of interpreting them as probabilities using equation 5.15. Admittedly, the interpretation of the direct γ coefficients as probabilities is not at all rigorous in this case, as a basis of only four chromosome profiles is used, but due to the larger number of orthogonal chromosome profiles used, the γ coefficients calculated using orthogonal chromosomes can be interpreted as probabilities with more confidence. This interpretation would allow one to set a probability threshold, with chromosomes which have their largest probability of belonging to a class below this threshold left unclassified. As an illustration, Figure 7.3 is a scatter plot of probabilities calculated using both types of γ coefficient, with chromosomes classified correctly and incorrectly indicated by the shape of the plotted point. Chromosomes were only classified if all three coefficients suggested the same class.

It can be seen on the graph that a large percentage of correctly classified chromosomes are clustered towards the top right of the plot, and that most of the incorrectly classified chromosomes appear in the less densely populated parts of the plot. Unfortunately, there is a large overlap between correctly and incorrectly classified chromosomes, so setting a threshold based purely on this information would exclude many correctly classified chromosomes too. Other information, such as the correlation coefficients would have to be considered as well.

Use of the sum rules introduced in Section 5.4.1 could also be useful in determining chromosomes which have been classified correctly, or in improving the quality of the γ coefficients calculated using orthogonal chromosomes. An example of the values of these coefficients for correctly and incorrectly classified chromosomes is provided in Figure 7.4, which shows a plot of $\sum \frac{c_j^2}{\omega_j}$ against $\sum c_j^2$, each sum being over 12 terms. It is clear that the values of $\sum c_j^2$ mostly cluster close to 1, and never exceed it, although a significant proportion of the values of the other sum exceed 1, often by large amounts. Examining this plot does not suggest any possible thresholds which can be set in order to separate

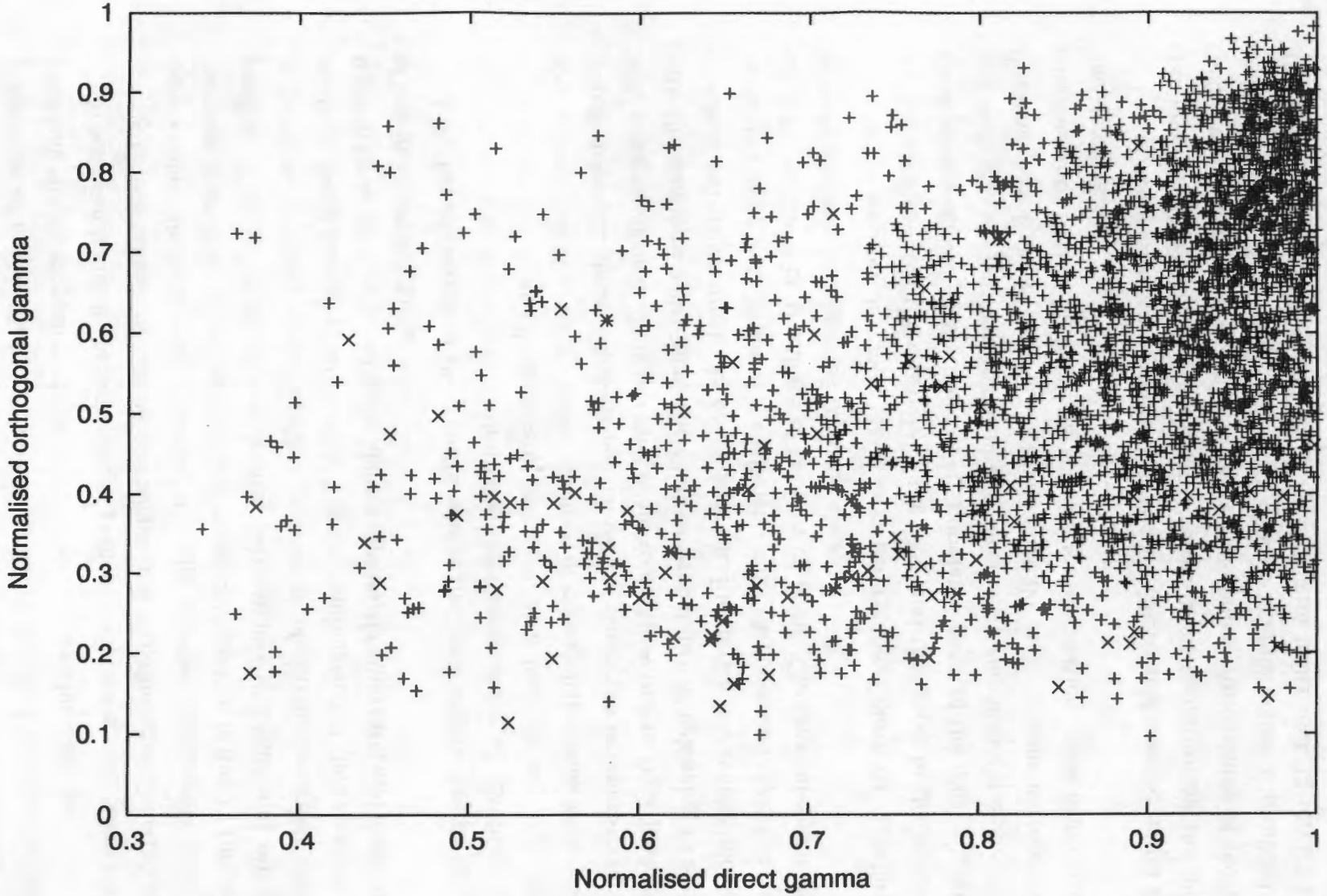


Figure 7.3: A scatter plot of the normalised γ coefficients calculated using orthogonal chromosomes (y-axis) against normalised direct γ coefficients (x-axis). Chromosomes classified correctly are marked $+$, and chromosomes classified incorrectly are marked $\times$. The plot is for the Cph data set, with average chromosomes calculated using part A of the data set, and the classification done on part B.

the correctly and incorrectly classified chromosomes. In order to test whether the position where $\sum \frac{c_i^2}{\omega_j}$ exceeds a certain threshold can be used as an indicator of where to truncate equation 5.28, which calculates values for the γ_j coefficients from the c_j coefficients, an experiment was done in which equation 5.28 was truncated one term before $\sum \frac{c_i^2}{\omega_j}$ exceeded 5 (an arbitrarily chosen value). Unfortunately, this was found to lead to slightly worse classification results.

As yet, no detailed investigation has been done into the use of various values which can be calculated as indicators of whether a classification is correct or not. Possible suggestions would be to look at correlations between correctly and incorrectly classified chromosomes with values of a large number of indicators, such as the sum rules, largest and smallest direct γ coefficients and values of the probabilities used simultaneously.

This classification based on profile information can be very useful in combination with another classifier using the features in Table 3.2 as inputs, as these features take profile information into consideration in a very limited way via the use of the weighted density distribution functions, but do take important features such as centromeric index into consideration. A useful model for classification might be to use a minimum Mahalanobis distance classifier operating on extracted features to replace the initial classification done by length-constrained correlation coefficients. The average chromosomes to use as bases for the calculation of γ coefficients can be chosen by using the chromosomes corresponding to classes with the lowest values of the Mahalanobis distance. Once all the profile coefficients have been calculated, chromosomes can be assigned to classes in a context-sensitive way by first assigning those chromosomes for which all four calculated coefficients agree, followed by those for which various triplets of coefficients agree and so on. The magnitudes of the probabilities estimated using the γ coefficients can also be taken into account within each group by assigning the chromosomes to classes in decreasing order of a probability value.

7.4 Comparison with published results

The only reported previous work in which classification decisions were made based on the correlation between the profiles of unknown chromosomes and a set of averaged chromosome profiles was done by Forabosco et al. [12]. The technique was tested on a set of 20 metaphases and a classification error of 18.2% was obtained. As a non-standard chromosome database was used, this result is unfortunately not directly comparable to results obtained in this work, although it falls into the same range as the results obtained using correlation coefficients and length-constrained correlation coefficients.

Much of the early work on classifying chromosomes, such as that done by Granlund [20] was based on classification of chromosomes using only profile information. Later techniques which use only profile information are some of the neural network experiments performed by Errington and Graham which only used a sampled profile and length as inputs to the

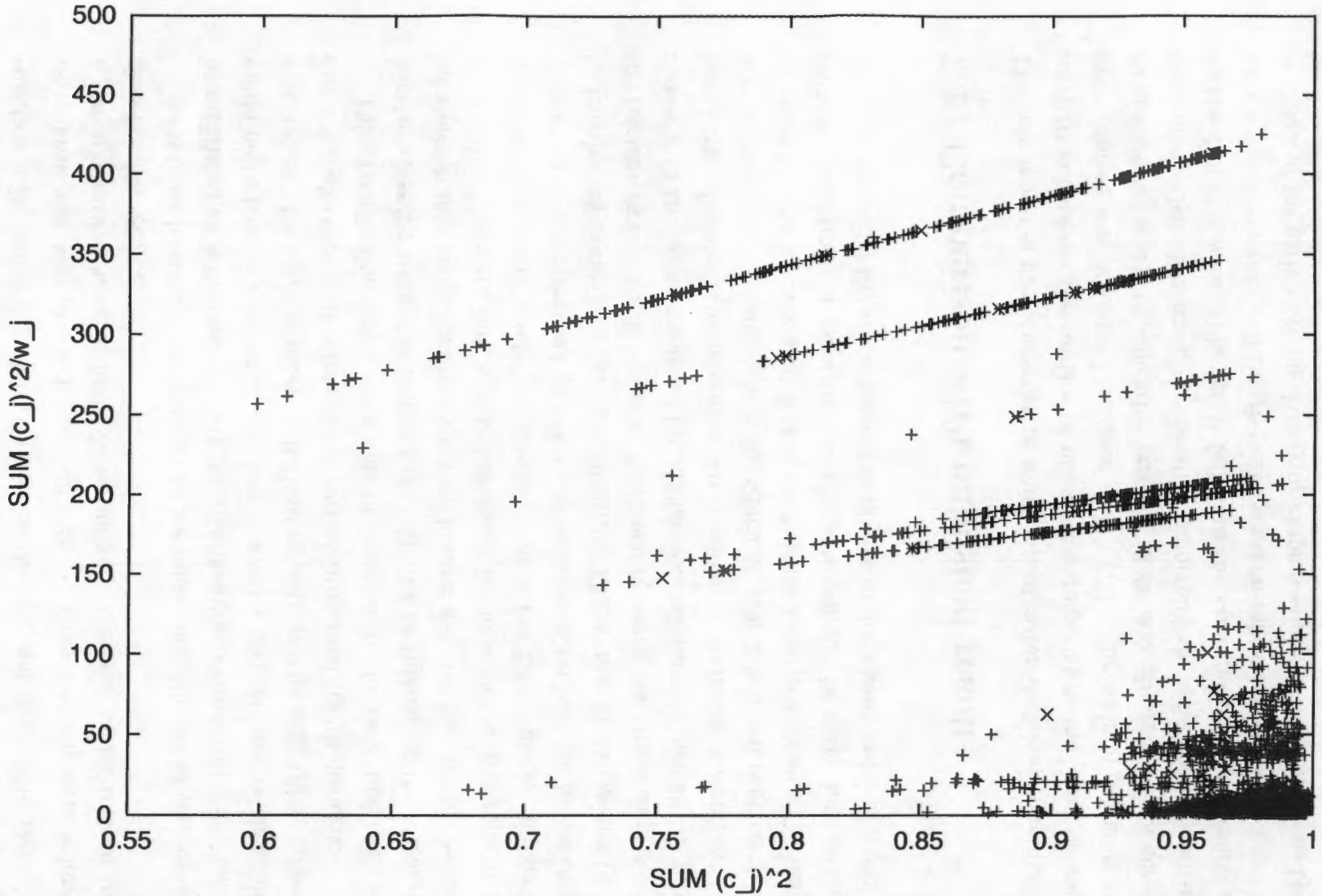


Figure 7.4: A scatter plot of the value of $\sum \frac{c_i^2}{\omega_j}$ against $\sum c_j^2$, with each sum being over 12 terms, for a number of correctly and incorrectly classified chromosomes. Chromosomes classified correctly are marked $+$, and chromosomes classified incorrectly are marked $\times$. The plot is for the Cph data set, with average chromosomes calculated using part A of the data set, and the classification done on part B.

Author	Method	Data Set	%MIS	%UN
Hanbury	Three profile coefficients	Cph	1.75	34.24
Hanbury	Three profile coefficients	Cpr	2.36	51.44
Granlund [20]	Curve matching	unknown	17.1	-
Granlund [20]	Sampled curves	unknown	15.0	-
Granlund [20]	Fourier descriptors	unknown	22.4	-
Granlund [20]	Distribution functions	unknown	9.9	-
Forabosco et al. [12]	Correlation coefficients	unknown	18.2	-
Errington et al. [10]	Neural network	Cph	8.4	-
Granum et al. [22]	Markov network models	Edited Cph	6.4	-
Johnston et al. [33]	Local band description	Edited Cph	30.5	-

Table 7.3: Comparison of chromosome classification techniques which only use profile and possibly length information for classification. The columns headed "%MIS" and "%UN" contain the percentage of chromosomes misclassified and the percentage of chromosomes left unclassified by each method. The top two rows present the results, described in this chapter, of using a combination of correlation coefficients, direct γ coefficients and γ coefficients calculated using orthogonal chromosomes to make a decision. The four Granlund techniques are described in Section 4.3.1; the technique used by Forabosco et al. is described in Section 4.3.3; the neural network result is obtained by feeding 15 sampled points from a profile and a normalised length to a 16-100-24 neural network, as described in Section 4.3.5; the Markov network models are described in Section 4.3.4; and the local band description technique is described in Section 4.3.6.

neural network [10] [11]; the Markov network models used by Granum and Thomason [22]; and the local band description technique used by Johnston, Tang and Zimmerman [33]. The results of using a combination of correlation coefficients, direct γ coefficients and γ coefficients calculated using orthogonal chromosomes are presented in Table 7.3, along with the performance statistics of other classifiers reported in the literature which only use profile and length information. Strictly, comparisons should not be made between these results due to the different data sets and experimental techniques used, although one can get a general idea of the performance of the techniques. Some results of experiments utilising banding patterns are not shown in the table due to the inclusion of centromeric index information as well as banding information in the features passed to the classifier. The results excluded are those of experiments performed by Lundsteen et al. [45] using band transition sequences to describe chromosomes (Section 4.3.2); the results of neural network experiments including centromeric index in the inputs (Section 4.3.5); and the local band description techniques used by Groen et al. [24] (Section 4.3.6).

It is clear from the table that the use of the combination of length-constrained correlation

coefficients, directly calculated γ coefficients and γ coefficients calculated using orthogonal chromosomes leads to a significantly smaller percentage of misclassified chromosomes than other profile-based methods.

Chapter 8

Conclusion

Advances in personal computer hardware now allow practical implementation of non-linear classification models in the form of neural networks. Neural networks are already used with great success in applications such as automated visual inspection of components on industrial production lines and in optical character recognition. As automated classification of chromosomes is a highly complicated problem, the use of a non-linear classification model would appear to be a good approach to improving the results. Unfortunately, the application of neural networks to chromosome classification proved to be disappointing, as is shown in Chapter 6. It was found that using efficient non-linear optimisation techniques to train neural networks having a sound theoretical basis for performing well as classifiers led to bad over-fitting of the neural network to the training set. The best solution would be to use a larger training set, but despite the speed of modern personal computers, the training time of a network on a large enough training set would still be prohibitive, and hence this solution is impractical at present.

Another technique often used in industrial inspection applications in order to find patterns in two-dimensional images is normalised grey-scale correlation. This technique is rather slow when applied to two-dimensional data, even when the fast Fourier transform method is used, but is tremendously fast when applied to one-dimensional data. An attempt at using normalised grey-scale correlation to classify chromosomes using integrated density profiles was made. Averages of the profiles of each of the 24 classes of chromosome were calculated, and an unknown chromosome was classified by assigning it to the class with the average profile having the highest correlation with the profile of the unknown chromosome. This technique, although simple, gave surprisingly good results.

An analysis shown in Chapter 5 demonstrates the difficulty of classifying chromosomes based on profiles alone by showing the large similarity (high correlation coefficients) between the averaged chromosome profiles in different classes. Borrowing some ideas from quantum mechanics led to attempts to represent a chromosome profile as a linear combination of average chromosome profiles, and hence determining the average profile to which it had

the largest resemblance based on the values of the expansion coefficients. Expansion of the unknown profile in terms of two average profiles was found to give no extra classification information above that gained from examining correlation coefficients. Eventually, the expansion was done in terms of four average profiles. The use of more than two chromosomes provides some information not available from the correlation coefficients due to the inclusion of more off-diagonal elements in the correlation matrix. However, the use of a larger basis proved to be unworkable, as this leads to large ill-conditioned systems of linear equations which must be solved.

Expansion of an unknown chromosome profile in terms of averages of real profiles has limitations, as these average profiles are not orthogonal to each other. This means that adding extra average profiles into the expansion does not necessarily improve the series representation of the profile, as it is only for a series expansion in terms of orthogonal functions that the accuracy of the series representation is guaranteed to increase as one adds more terms. Usually, orthogonal functions used in series expansions are determined from theoretical models of the function being expanded, but as no such models of chromosome profiles exist, a set of orthogonal chromosome profiles was constructed from the averaged chromosome profiles. Because of the unusual way in which these orthogonal profiles were constructed using averages of "experimental" data, it was not possible to use all twenty-four orthogonal profiles, as the later orthogonal profiles were found to be contaminated by noise. It was therefore decided to represent real chromosome profiles as linear expansions of twelve orthogonal chromosome profiles. The expansion coefficients in terms of orthogonal profiles were then converted back to expansion coefficients in terms of real chromosome profiles by a linear transformation, and these coefficients were used in making classification decisions. Due to the larger basis used in this case, one has some justification in interpreting the squares of the expansion coefficients as probabilities in a quantum mechanical sense, where these probabilities have the advantage that no underlying parametric model is assumed. This interpretation suggests a very simple procedure for judging the reliability of a classification based only on the chromosome banding profiles.

It was found that using a combination of correlation coefficients, expansion coefficients of real profiles in terms of four average profiles, and expansion coefficients of real profiles calculated using twelve orthogonal profiles, where an unknown chromosome was only assigned to a class if the highest coefficient of all three sets agreed on the same class led to low misclassification rates. These were lower than misclassification rates obtained by using other techniques reported in the literature which classified chromosomes based only on banding patterns. This technique does leave a large number of chromosomes unclassified, which is slightly disappointing, but this is felt to be tolerable given the difficulty of assigning chromosomes to classes based only on profile information.

The use of the techniques developed in this dissertation allows a large amount of information to be extracted from chromosome profiles, but does not take other pertinent

features such as centromeric index and chromosome area into account. At present, these techniques are competitive with, but not clearly superior to existing techniques. Making the best use of these techniques would most likely involve combining them with a classifier which can make use of the non-profile features, such as a minimum Mahalanobis distance classifier, and developing a method by which the probabilities of chromosomes belonging to classes suggested by the two methods can be combined.

Appendix A

Normalised Greyscale Correlation using the Fast Fourier Transform

The correlation of two continuous functions $g(x)$ and $h(x)$ is defined as [66]

$$\text{Corr}(g,h) \equiv \int_{-\infty}^{\infty} g(r+x)\, h(r)\, dr \tag{A.1}$$

which is a function of x (the lag). The correlation will be large at some value of x if $g(x)$ is very similar to $h(x)$, but is shifted to the right by x. Similarly, the correlation will be large for some negative value of x if g is shifted to the left of h. Equation A.1 is a member of the Fourier transform pair

$$\text{Corr}(g,h) \Leftrightarrow G(f)\, H^*(f) \tag{A.2}$$

where $G(f)$ and $H(f)$ are the Fourier transforms of $g(x)$ and $h(x)$ respectively, and the asterisk denotes complex conjugation.

Now consider two discretely sampled function g_k and h_k which are both periodic with period N. The discrete correlation of these two functions is defined by

$$\text{Corr}(g,h)_j \equiv \sum_{k=0}^{N-1} g_{j+k} h_k \tag{A.3}$$

which is a member of the discrete Fourier transform pair

$$\text{Corr}(g,h)_j \Leftrightarrow G_k H_k^* \tag{A.4}$$

where G_k and H_k are the Fourier transforms of g_j and h_j respectively.

To compute the correlation of two discretely sampled real functions, one simply has to Fourier transform each function, multiply one transform by the complex conjugate of the other, and inverse transform the result. The resulting vector $\mathbf{C}$ is real (as both the initial functions were real), and contains the correlation coefficients at different lags.

In order to obtain the position of maximum correlation with subpixel accuracy, the element m of $\mathbf{C}$ with the largest correlation coefficient is found, a parabola is fitted exactly to the correlation coefficients $\mathbf{C}_{m-1}$, $\mathbf{C}_m$ and $\mathbf{C}_{m+1}$, and the position of maximum correlation is taken to be at the maximum of the fitted parabola.

Appendix B

Chromosome Statistics

This section presents the profile length statistics for chromosomes in the Copenhagen (Table B.1), Edinburgh (Table B.2), Philadelphia (Table B.3) and Cpr (Table B.4) data sets. The number of chromosomes in each class, the mean and standard deviations of the lengths and the minimum and maximum length per class is given. The lengths quoted are the number of fixed length sampling steps required to sample the whole profile.

Denver Class	Chromosome Class	Count	Length Statistics			
			Mean	SD	Min	Max
A	1	342	87	13	60	127
	2	341	83	12	59	130
	3	343	70	10	41	99
B	4	346	66	9	43	93
	5	346	64	9	44	95
C	6	348	62	8	44	86
	7	351	57	8	32	86
	8	351	52	7	36	96
	9	351	50	6	35	66
	10	355	49	7	17	70
	11	349	49	6	35	68
	12	351	49	6	35	69
	23 (X)	262	54	8	24	79
D	13	354	42	6	26	61
	14	356	41	5	28	64
	15	355	40	5	28	55
E	16	354	35	4	26	51
	17	360	35	4	26	50
	18	360	33	4	23	46
F	19	360	29	4	18	41
	20	360	29	4	21	43
G	21	360	23	3	14	37
	22	360	26	3	17	39
	24 (Y)	91	27	3	20	34

Table B.1: The length statistics of the 8106 chromosomes in the Copenhagen (Cph) data set. The chromosomes are grouped by Denver class. The numbers in the "Count" column show the number of chromosomes of each class in the data set. The last four columns give the mean and standard deviation (SD) of the lengths, and the minimum and maximum lengths for each class of chromosome.

Denver Class	Chromosome Class	Count	Length Statistics			
			Mean	SD	Min	Max
A	1	242	66	10	31	93
	2	240	64	9	34	90
	3	242	54	7	33	79
B	4	246	50	7	31	72
	5	242	48	7	27	67
C	6	243	47	7	25	69
	7	245	43	6	27	58
	8	240	39	5	24	53
	9	241	38	5	22	55
	10	245	37	5	23	51
	11	240	37	5	24	53
	12	239	36	5	22	54
	23 (X)	122	43	5	30	58
D	13	240	31	4	20	43
	14	242	29	4	17	47
	15	244	29	4	19	42
E	16	238	25	4	15	35
	17	240	24	3	15	33
	18	244	24	3	15	37
F	19	232	18	4	9	27
	20	241	20	2	8	24
G	21	243	14	3	8	24
	22	237	16	3	9	25
	24 (Y)	120	18	3	13	28

Table B.2: The length statistics of the 5548 chromosomes in the Edinburgh (Edi) data set. The chromosomes are grouped by Denver class. The numbers in the "Count" column show the number of chromosomes of each class in the data set. The last four columns give the mean and standard deviation (SD) of the lengths, and the minimum and maximum lengths for each class of chromosome.

Denver Class	Chromosome Class	Count	Length Statistics			
			Mean	SD	Min	Max
A	1	258	71	13	37	139
	2	259	67	12	40	131
	3	259	56	9	31	86
B	4	257	52	8	34	102
	5	258	51	9	29	88
C	6	259	49	8	31	76
	7	258	45	7	30	66
	8	259	41	7	25	67
	9	258	40	6	25	64
	10	258	39	6	26	60
	11	257	40	6	25	59
	12	258	39	6	26	62
	23 (X)	192	42	7	27	62
D	13	259	33	6	21	53
	14	259	33	6	21	52
	15	258	32	5	20	53
E	16	261	29	5	14	43
	17	258	28	5	18	51
	18	259	26	4	18	43
F	19	261	23	4	13	38
	20	260	23	4	15	36
G	21	258	20	4	12	36
	22	257	21	4	13	36
	24 (Y)	65	22	4	14	33

Table B.3: The length statistics of the 5945 chromosomes in the Philadelphia (Phi) data set. The chromosomes are grouped by Denver class. The numbers in the "Count" column show the number of chromosomes of each class in the data set. The last four columns give the mean and standard deviation (SD) of the lengths, and the minimum and maximum lengths for each class of chromosome.

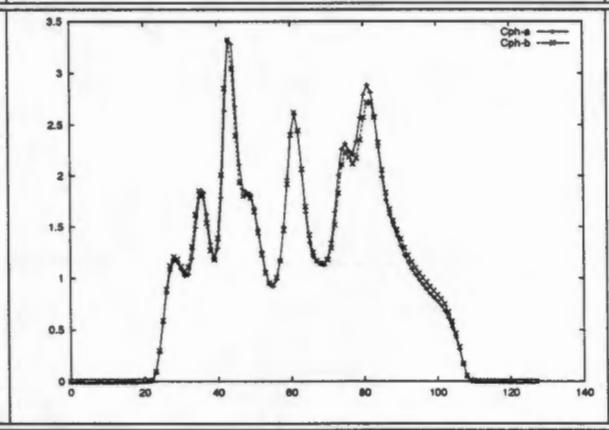
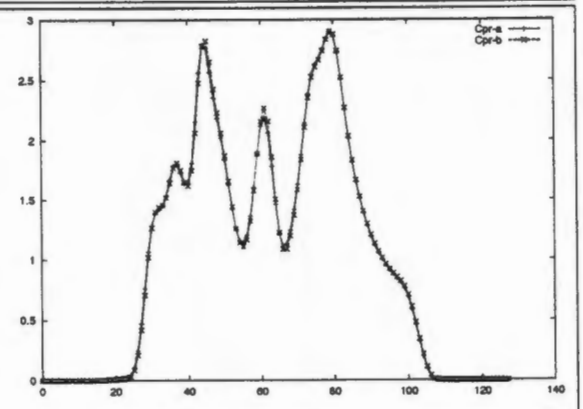
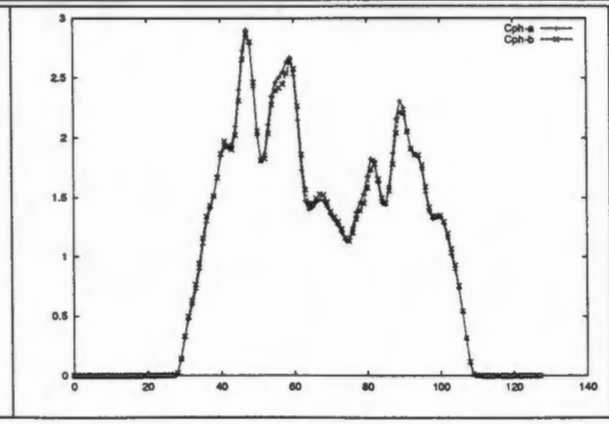
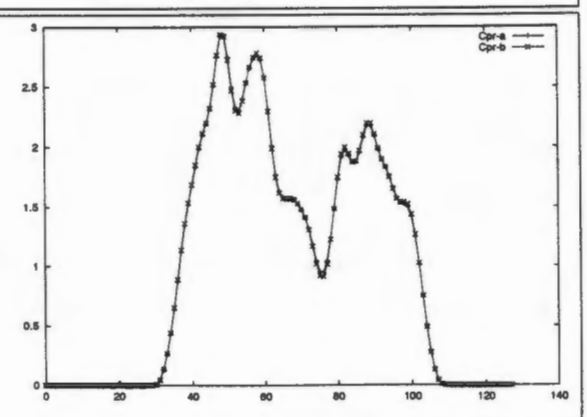
Denver Class	Chromosome Class	Count	Length Statistics			
			Mean	SD	Min	Max
A	1	5609	81	12	35	127
	2	5607	76	11	37	124
	3	5608	65	9	34	103
B	4	5607	60	8	33	97
	5	5608	59	8	29	94
C	6	5608	57	8	27	86
	7	5608	53	7	33	79
	8	5608	48	6	29	76
	9	5608	47	6	27	81
	10	5608	46	6	29	72
	11	5606	46	6	22	71
	12	5607	46	6	30	71
	23 (X)	4140	50	7	26	76
D	13	5606	39	5	23	69
	14	5601	38	5	24	62
	15	5609	39	5	25	58
E	16	5611	34	4	21	60
	17	5610	34	4	20	52
	18	5613	31	4	20	51
F	19	5608	28	4	17	47
	20	5607	28	3	19	42
G	21	5619	22	3	14	42
	22	5606	25	3	16	43
	24 (Y)	1468	25	3	17	62

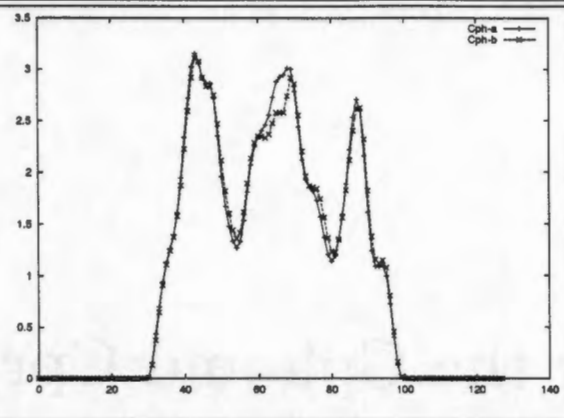
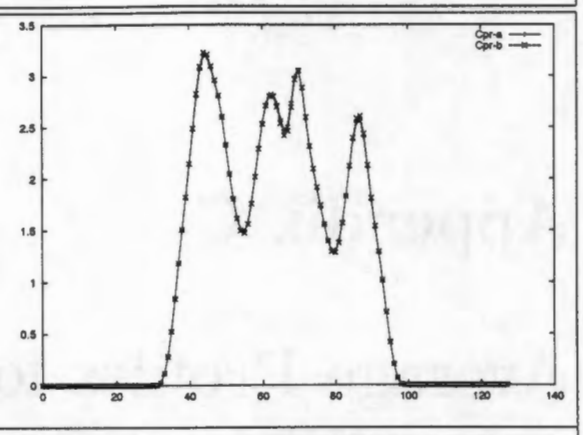
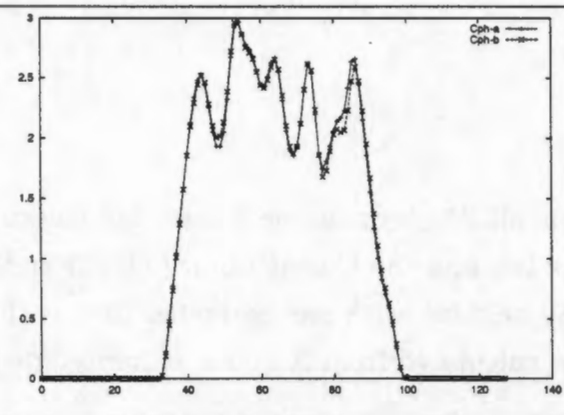
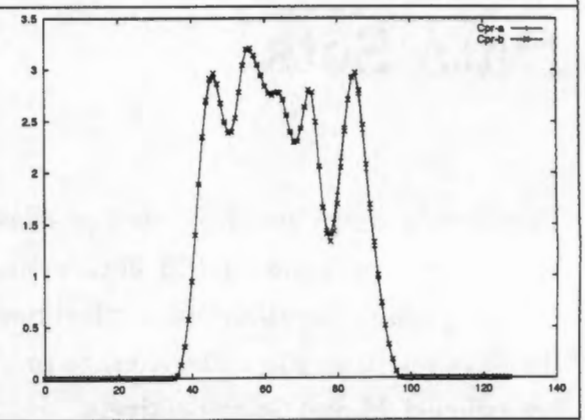
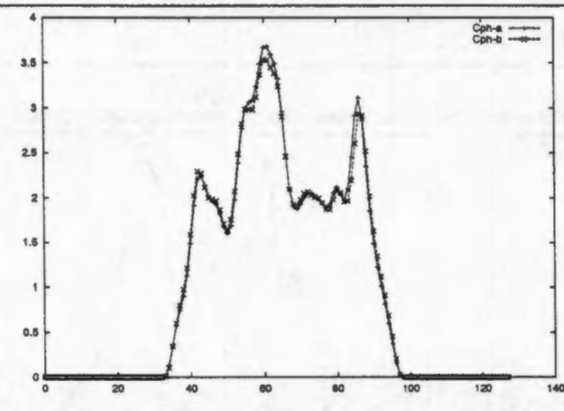
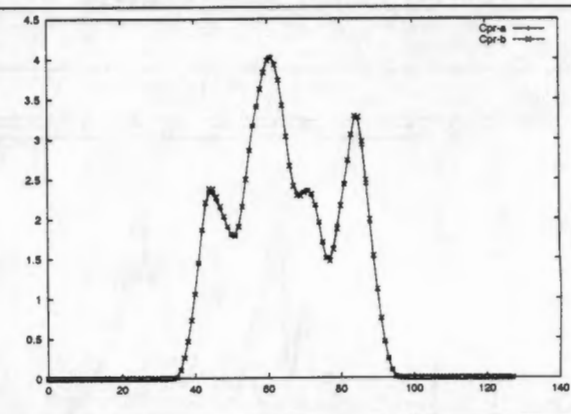
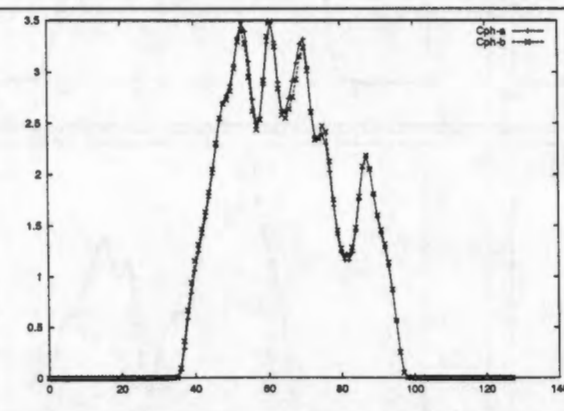
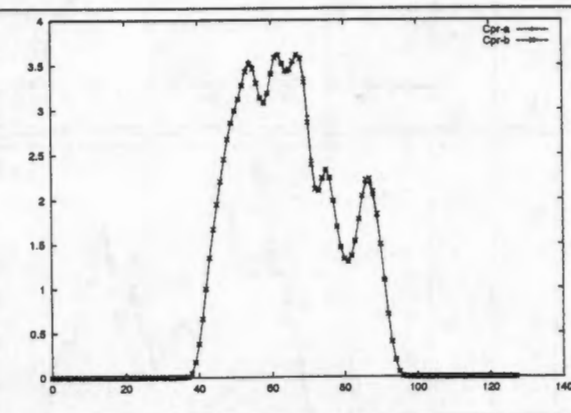
Table B.4: The length statistics of the 128990 chromosomes in the Cpr data set. The chromosomes are grouped by Denver class. The numbers in the "Count" column show the number of chromosomes of each class in the data set. The last four columns give the mean and standard deviation (SD) of the lengths, and the minimum and maximum lengths for each class of chromosome.

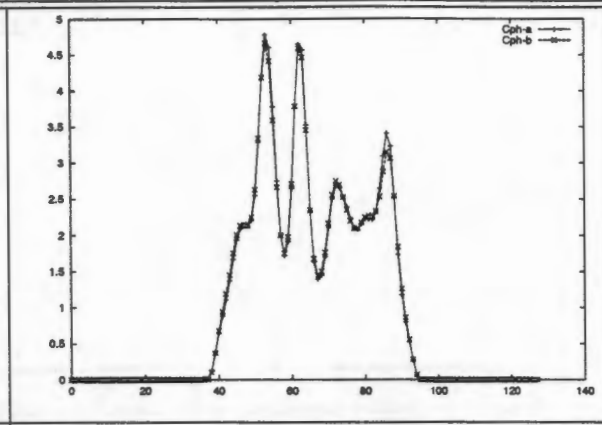
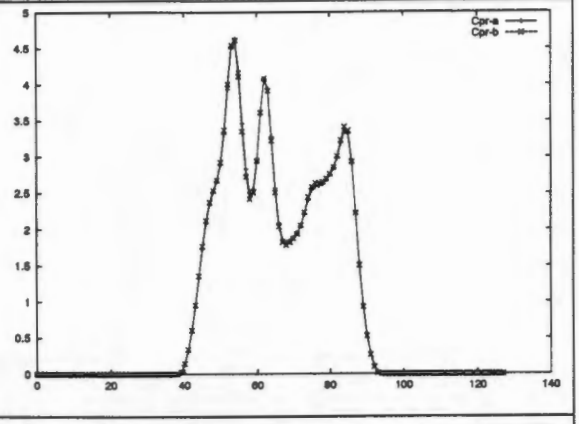
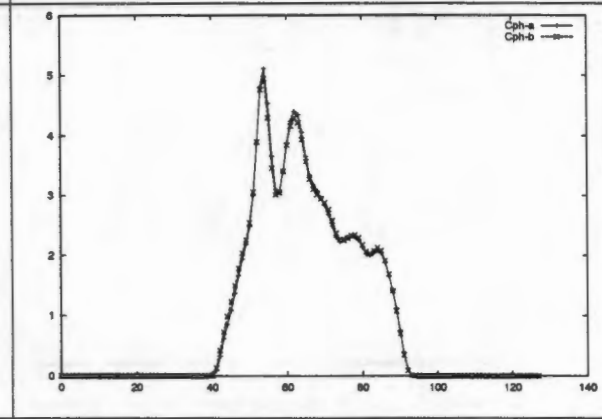
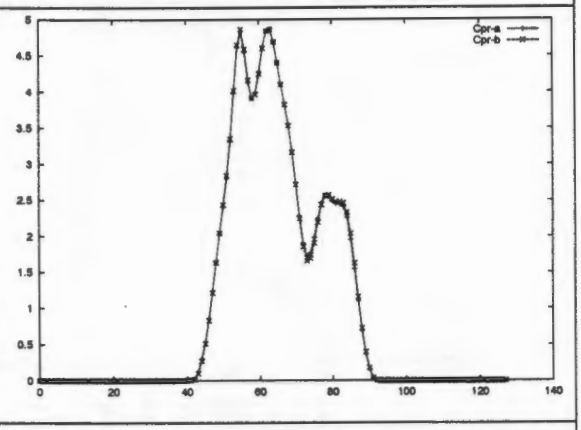
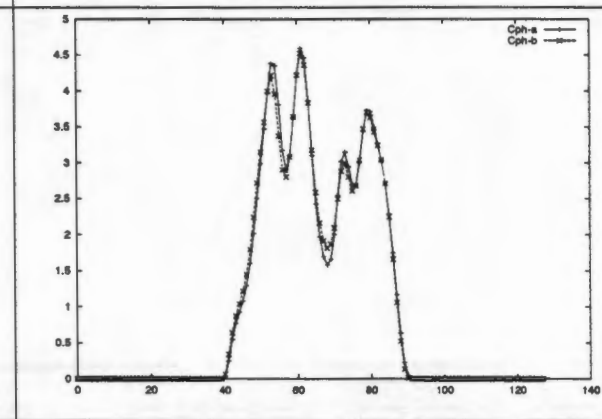
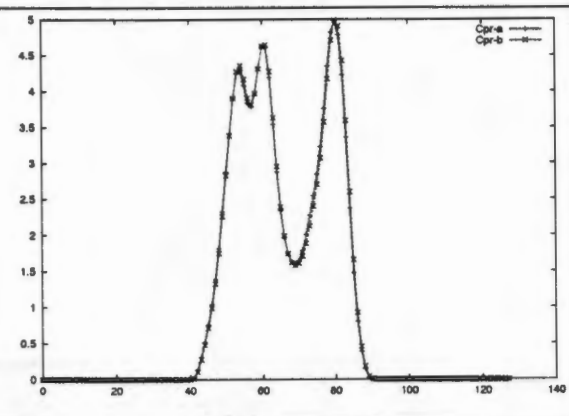
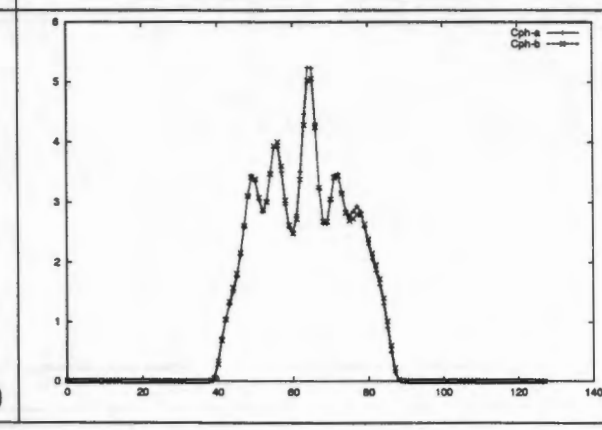
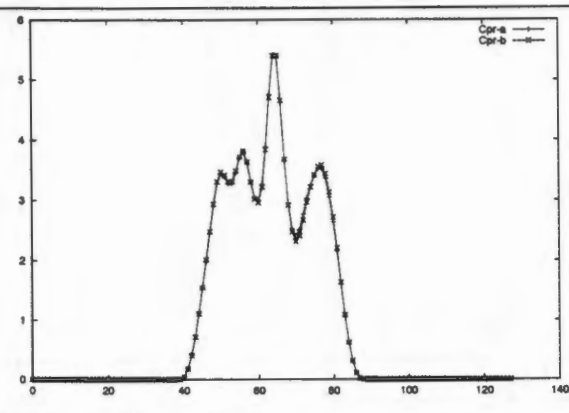
Appendix C

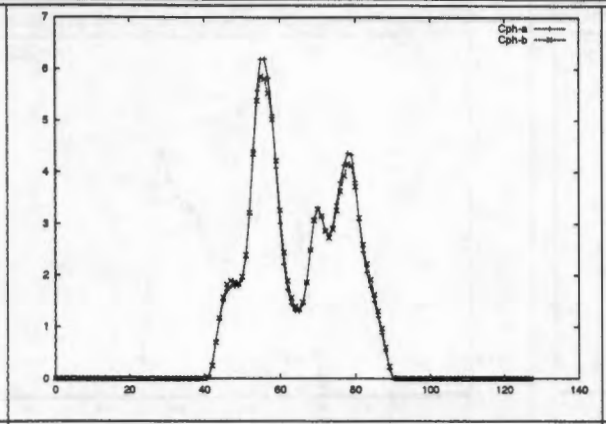
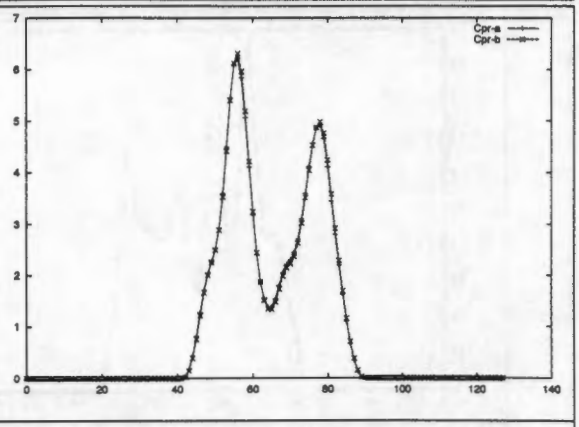
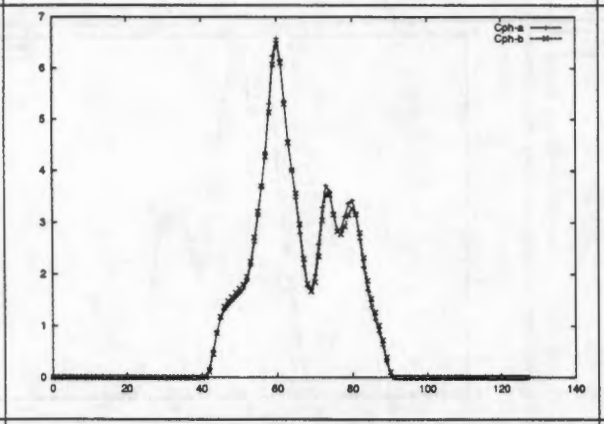
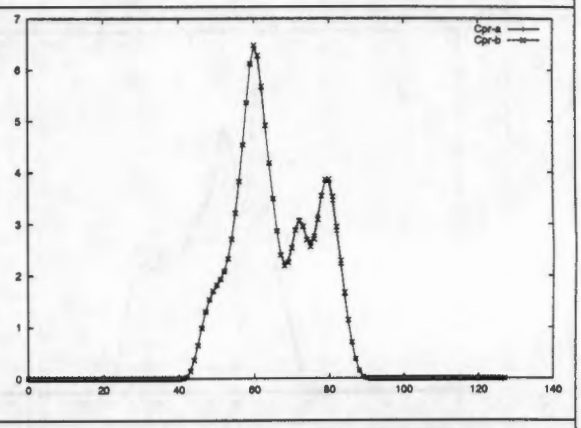
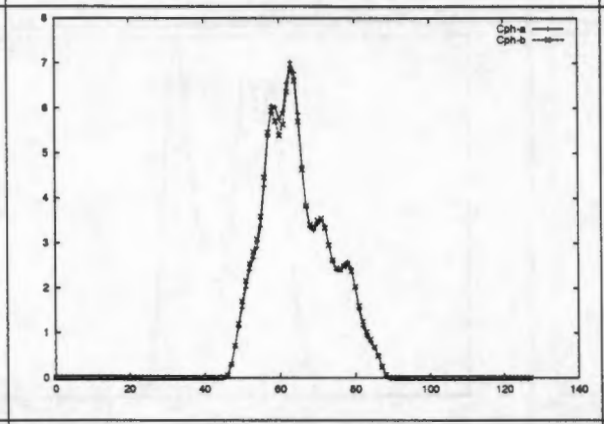
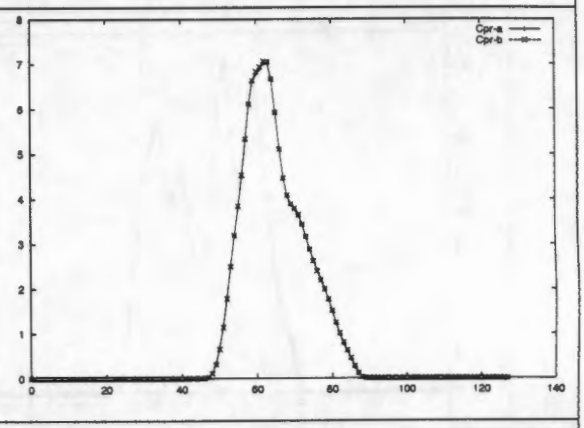
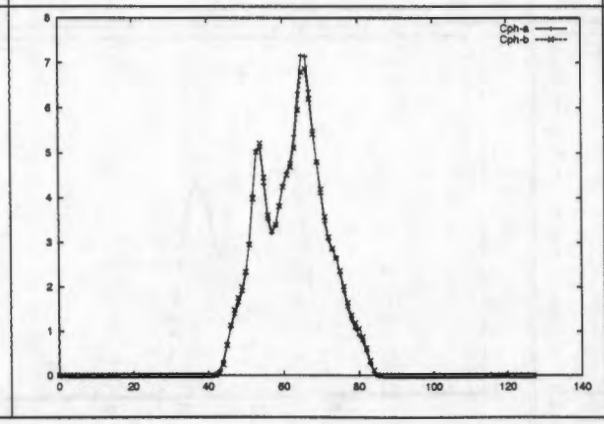
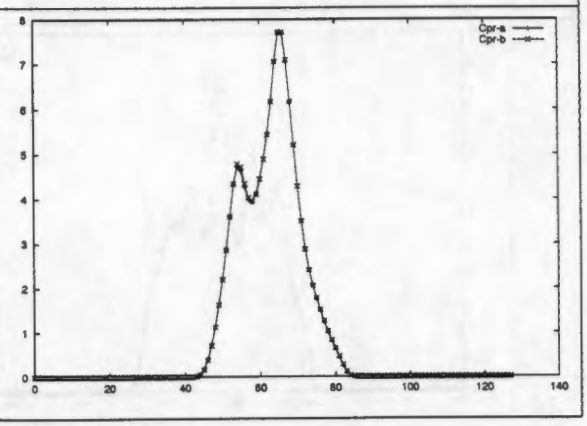
Average Profiles for the Cph and Cpr Data Sets

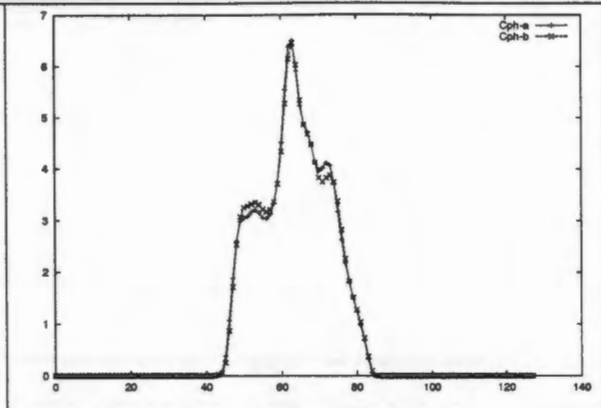
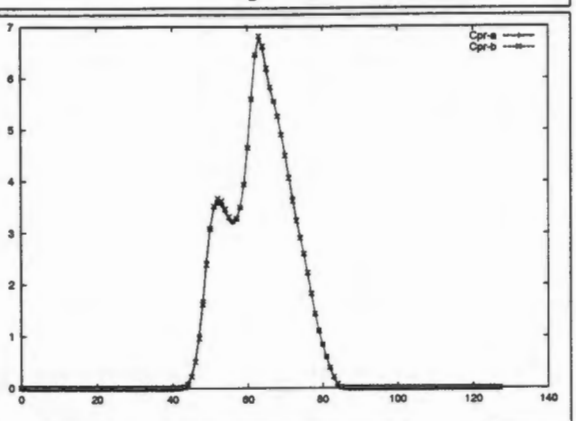
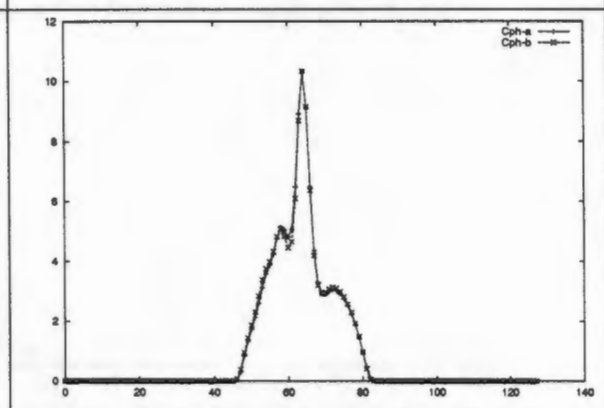
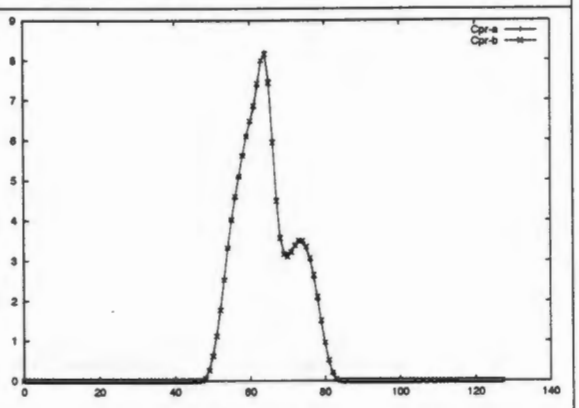
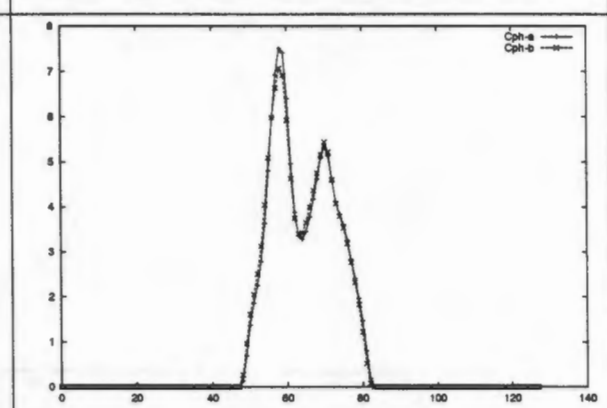
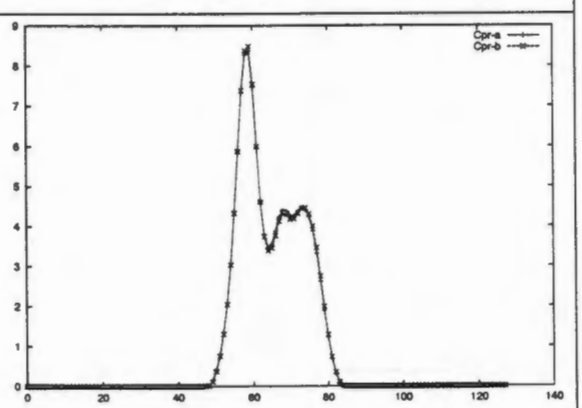
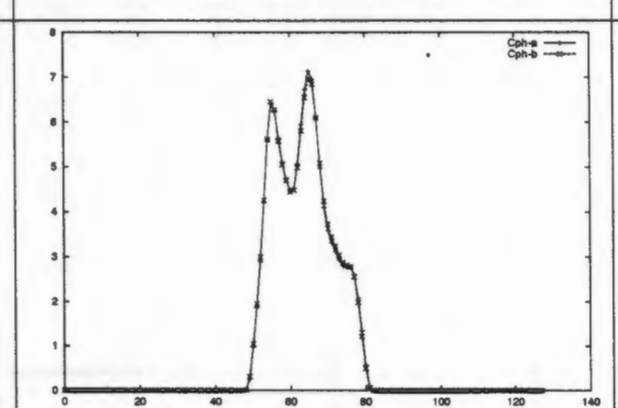
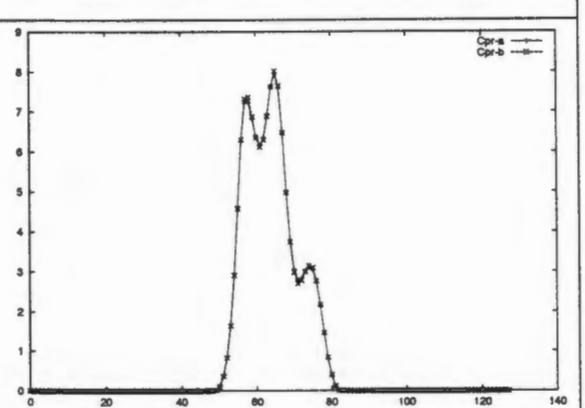
The Cph-M and Cpr-M average profiles for all 24 chromosome classes are shown in the table below, with the Cph-M library on the left and the Cpr-M library on the right. Two average profiles are shown for each chromosome class, with one generated from each half of the data set (a and b). The average profiles calculated from X and Y chromosome profiles are labelled 23 and 24 respectively.

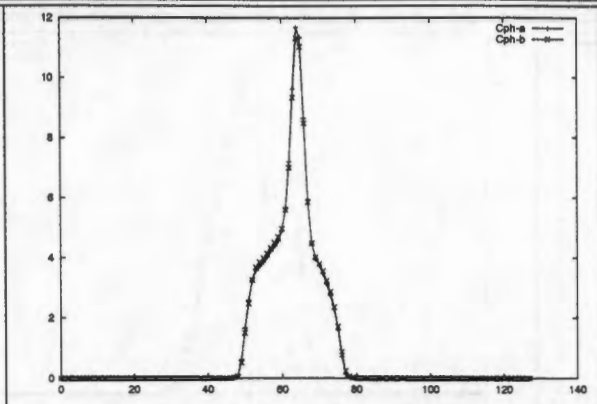
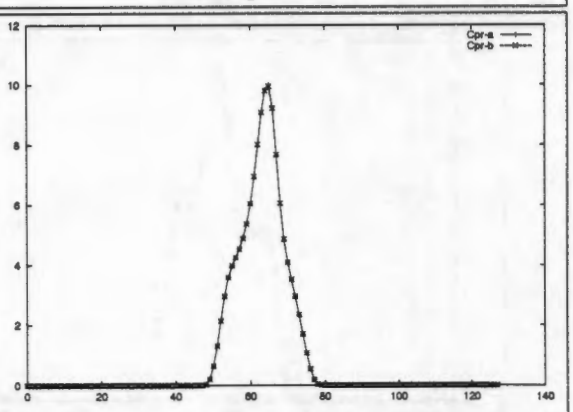
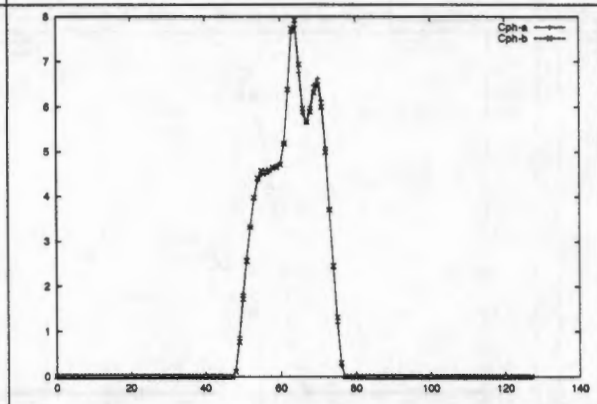
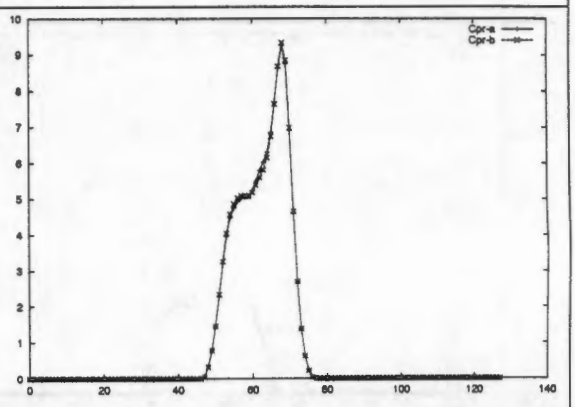
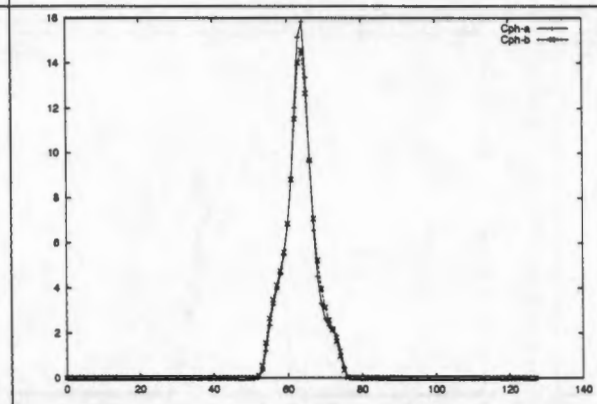
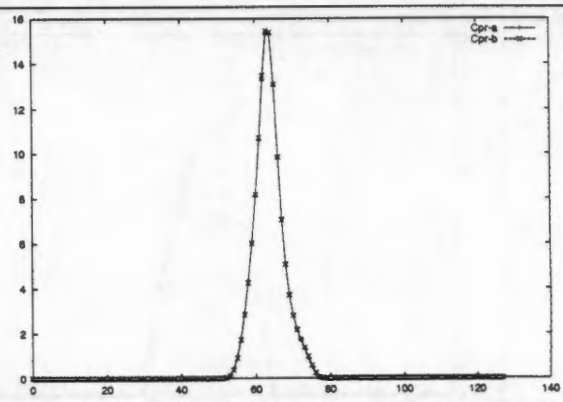
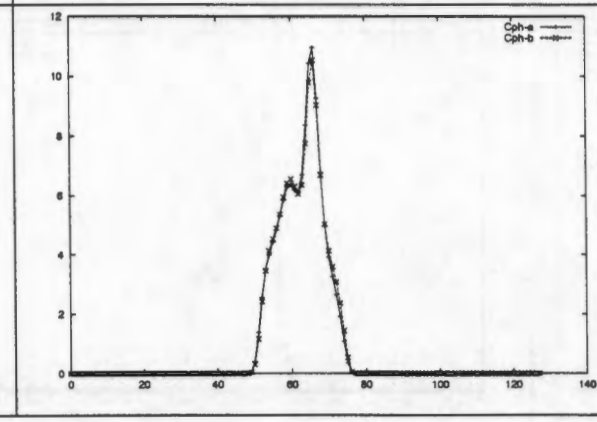
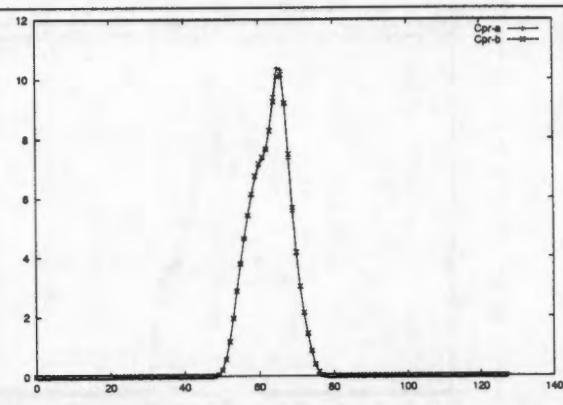
#	Cph-M	Cpr-M
1		
2		

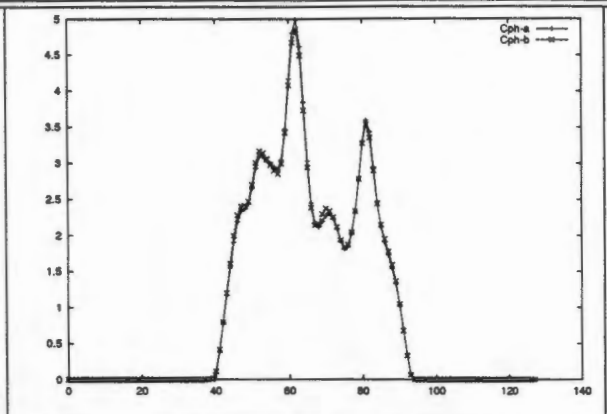
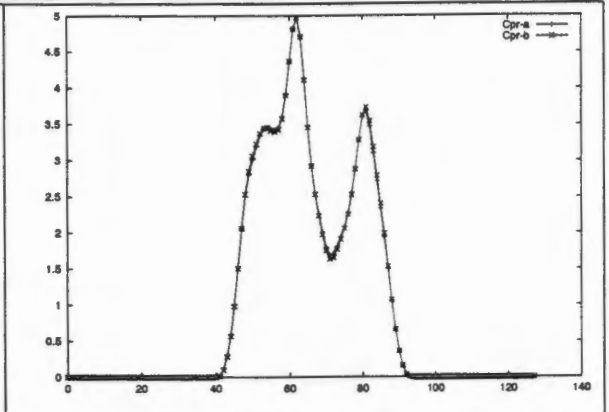
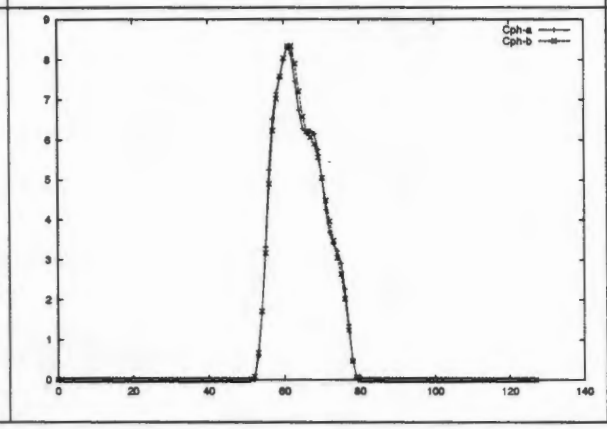
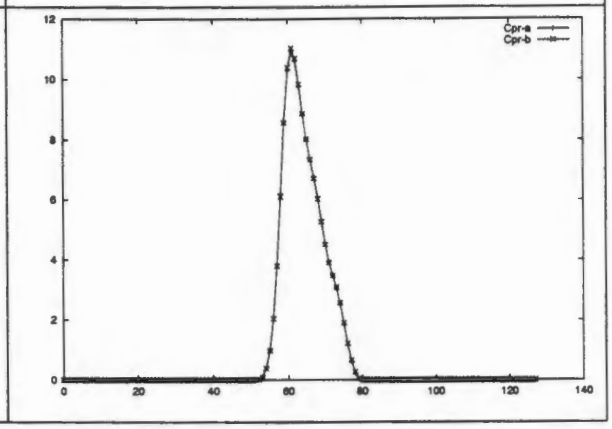
#	Cph-M	Cpr-M
3		
4		
5		
6		

#	Cph-M	Cpr-M
7		
8		
9		
10		

#	Cph-M	Cpr-M
11		
12		
13		
14		

#	Cph-M	Cpr-M
15		
16		
17		
18		

#	Cph-M	Cpr-M
19		
20		
21		
22		

#	Cph-M	Cpr-M
23		
24		

Appendix D

Library Correlation Coefficients

The overlaps (zero-shift correlation coefficients) of averaged chromosome profiles with other averaged profiles in the same library for the Cph-Ma (Table D.1), Cph-La (Table D.2), Cpr-Ma (Table D.3) and Cpr-La (Table D.4) libraries are presented in this appendix. The average profiles calculated from X and Y chromosome profiles are labelled 23 and 24 respectively. Only the correlation coefficients calculated from subset a of each data set are shown, due to the similarity of overlaps calculated using each subset. As the matrices are symmetrical, only the upper triangular section of each matrix is shown. As the correlation of a library profile with itself is always perfect, the matrix diagonal elements are all ones.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.899	0.894	0.896	0.874	0.813	0.806	0.739	0.789	0.746	0.701	0.748	0.621	0.572	0.634	0.545	0.568	0.526	0.468	0.492	0.411	0.441	0.824	0.504
2		1.000	0.907	0.917	0.893	0.880	0.821	0.784	0.771	0.762	0.745	0.730	0.645	0.640	0.660	0.580	0.631	0.604	0.521	0.565	0.420	0.526	0.822	0.544
3			1.000	0.949	0.931	0.921	0.844	0.809	0.765	0.821	0.699	0.745	0.704	0.727	0.750	0.666	0.666	0.666	0.634	0.672	0.552	0.629	0.841	0.645
4				1.000	0.982	0.957	0.943	0.908	0.897	0.886	0.843	0.844	0.775	0.769	0.789	0.719	0.742	0.738	0.660	0.701	0.540	0.641	0.926	0.656
5					1.000	0.952	0.944	0.927	0.909	0.887	0.838	0.897	0.840	0.795	0.821	0.779	0.785	0.783	0.720	0.744	0.637	0.706	0.951	0.741
6						1.000	0.944	0.952	0.906	0.925	0.854	0.862	0.836	0.867	0.887	0.786	0.823	0.817	0.739	0.807	0.602	0.729	0.935	0.745
7							1.000	0.954	0.947	0.892	0.844	0.857	0.800	0.797	0.829	0.753	0.728	0.756	0.695	0.732	0.577	0.637	0.955	0.650
8								1.000	0.963	0.950	0.899	0.921	0.913	0.921	0.925	0.869	0.866	0.898	0.820	0.860	0.691	0.802	0.965	0.798
9									1.000	0.925	0.918	0.943	0.868	0.822	0.867	0.789	0.818	0.799	0.706	0.749	0.586	0.671	0.972	0.710
10										1.000	0.876	0.899	0.904	0.920	0.942	0.888	0.850	0.888	0.831	0.851	0.706	0.792	0.929	0.786
11											1.000	0.872	0.800	0.763	0.777	0.702	0.865	0.796	0.596	0.695	0.424	0.614	0.864	0.666
12												1.000	0.945	0.837	0.898	0.863	0.902	0.852	0.779	0.803	0.715	0.766	0.941	0.858
13													1.000	0.930	0.955	0.968	0.935	0.952	0.923	0.931	0.862	0.906	0.884	0.948
14														1.000	0.966	0.944	0.881	0.967	0.942	0.958	0.820	0.942	0.840	0.886
15															1.000	0.944	0.898	0.931	0.912	0.942	0.820	0.880	0.881	0.903
16																1.000	0.863	0.957	0.978	0.935	0.924	0.931	0.816	0.919
17																	1.000	0.927	0.803	0.896	0.678	0.839	0.801	0.920
18																		1.000	0.935	0.948	0.825	0.944	0.803	0.920
19																			1.000	0.949	0.950	0.962	0.755	0.904
20																				1.000	0.853	0.940	0.781	0.930
21																					1.000	0.899	0.672	0.877
22																						1.000	0.725	0.928
23																							1.000	0.756
24																								1.000

Table D.1: CPH-Ma library correlation coefficients

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.918	0.905	0.923	0.891	0.911	0.933	0.928	0.924	0.907	0.889	0.859	0.879	0.903	0.925	0.838	0.940	0.912	0.858	0.916	0.796	0.936	0.901	0.924
2		1.000	0.924	0.969	0.944	0.943	0.928	0.959	0.966	0.943	0.941	0.929	0.925	0.871	0.904	0.853	0.962	0.927	0.824	0.926	0.794	0.890	0.967	0.960
3			1.000	0.949	0.933	0.966	0.920	0.943	0.920	0.953	0.841	0.838	0.876	0.933	0.935	0.907	0.911	0.977	0.916	0.950	0.857	0.901	0.924	0.942
4				1.000	0.979	0.963	0.943	0.953	0.957	0.961	0.925	0.941	0.937	0.909	0.954	0.906	0.956	0.955	0.891	0.960	0.862	0.932	0.973	0.972
5					1.000	0.952	0.937	0.950	0.959	0.944	0.865	0.960	0.933	0.890	0.959	0.923	0.909	0.940	0.875	0.945	0.897	0.905	0.976	0.946
6						1.000	0.948	0.982	0.946	0.964	0.892	0.913	0.958	0.955	0.949	0.933	0.957	0.980	0.920	0.941	0.910	0.928	0.953	0.985
7							1.000	0.957	0.976	0.894	0.904	0.898	0.921	0.871	0.905	0.855	0.934	0.913	0.824	0.907	0.818	0.878	0.949	0.933
8								1.000	0.968	0.947	0.901	0.929	0.973	0.937	0.944	0.924	0.956	0.966	0.876	0.923	0.898	0.904	0.970	0.979
9									1.000	0.923	0.918	0.945	0.931	0.869	0.914	0.868	0.942	0.913	0.816	0.926	0.821	0.882	0.976	0.937
10										1.000	0.896	0.904	0.907	0.956	0.957	0.947	0.950	0.969	0.939	0.977	0.891	0.944	0.928	0.959
11											1.000	0.884	0.890	0.834	0.831	0.782	0.971	0.847	0.787	0.893	0.710	0.868	0.900	0.913
12												1.000	0.953	0.850	0.914	0.873	0.908	0.871	0.812	0.891	0.868	0.888	0.959	0.925
13													1.000	0.907	0.924	0.907	0.938	0.928	0.842	0.870	0.911	0.884	0.957	0.972
14														1.000	0.959	0.944	0.917	0.963	0.977	0.943	0.932	0.956	0.887	0.940
15															1.000	0.958	0.903	0.966	0.945	0.957	0.945	0.953	0.944	0.946
16																1.000	0.863	0.954	0.922	0.911	0.973	0.878	0.891	0.917
17																	1.000	0.930	0.867	0.927	0.811	0.933	0.926	0.973
18																		1.000	0.934	0.940	0.921	0.922	0.934	0.976
19																			1.000	0.949	0.913	0.956	0.846	0.896
20																				1.000	0.858	0.957	0.929	0.923
21																					1.000	0.867	0.868	0.893
22																						1.000	0.889	0.930
23																							1.000	0.955
24																								1.000

Table D.2: CPH-La library correlation coefficients

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.883	0.899	0.866	0.847	0.796	0.814	0.713	0.777	0.753	0.737	0.739	0.595	0.543	0.607	0.564	0.592	0.516	0.457	0.455	0.379	0.426	0.784	0.461
2		1.000	0.912	0.911	0.885	0.872	0.844	0.775	0.763	0.758	0.731	0.734	0.635	0.634	0.660	0.600	0.617	0.582	0.548	0.575	0.430	0.529	0.814	0.502
3			1.000	0.953	0.931	0.914	0.850	0.791	0.735	0.806	0.690	0.755	0.699	0.706	0.749	0.675	0.661	0.646	0.638	0.655	0.543	0.613	0.817	0.615
4				1.000	0.980	0.964	0.950	0.893	0.856	0.878	0.826	0.848	0.771	0.766	0.794	0.740	0.745	0.716	0.682	0.697	0.547	0.652	0.915	0.627
5					1.000	0.953	0.943	0.922	0.879	0.874	0.811	0.902	0.838	0.791	0.819	0.803	0.791	0.779	0.744	0.733	0.648	0.725	0.943	0.723
6						1.000	0.947	0.950	0.869	0.934	0.840	0.883	0.854	0.880	0.900	0.834	0.814	0.812	0.794	0.818	0.652	0.766	0.938	0.727
7							1.000	0.942	0.941	0.907	0.891	0.887	0.785	0.773	0.808	0.758	0.735	0.706	0.676	0.677	0.550	0.630	0.971	0.604
8								1.000	0.933	0.949	0.882	0.948	0.927	0.920	0.927	0.905	0.868	0.882	0.858	0.858	0.728	0.833	0.973	0.786
9									1.000	0.906	0.944	0.948	0.816	0.750	0.790	0.779	0.797	0.728	0.658	0.654	0.539	0.625	0.968	0.624
10										1.000	0.880	0.917	0.885	0.902	0.929	0.887	0.837	0.852	0.820	0.797	0.698	0.781	0.934	0.740
11											1.000	0.881	0.759	0.710	0.734	0.724	0.823	0.705	0.578	0.624	0.381	0.548	0.883	0.508
12												1.000	0.939	0.838	0.881	0.908	0.916	0.872	0.803	0.770	0.718	0.784	0.955	0.821
13													1.000	0.938	0.951	0.991	0.948	0.976	0.944	0.902	0.863	0.930	0.867	0.939
14														1.000	0.981	0.943	0.855	0.940	0.968	0.967	0.840	0.950	0.828	0.874
15															1.000	0.951	0.871	0.923	0.945	0.934	0.841	0.914	0.863	0.892
16																1.000	0.924	0.981	0.958	0.886	0.891	0.937	0.838	0.937
17																	1.000	0.934	0.828	0.836	0.686	0.826	0.802	0.846
18																		1.000	0.958	0.909	0.867	0.958	0.794	0.925
19																			1.000	0.946	0.938	0.990	0.762	0.943
20																				1.000	0.805	0.948	0.741	0.870
21																					1.000	0.929	0.669	0.943
22																						1.000	0.732	0.946
23																							1.000	0.723
24																								1.000

Table D.3: CPR-Ma library correlation coefficients

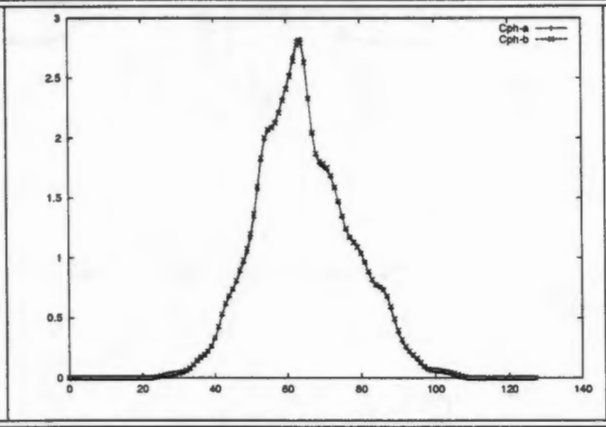
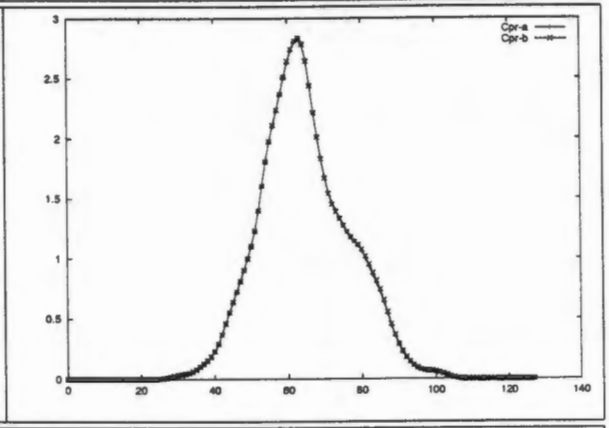
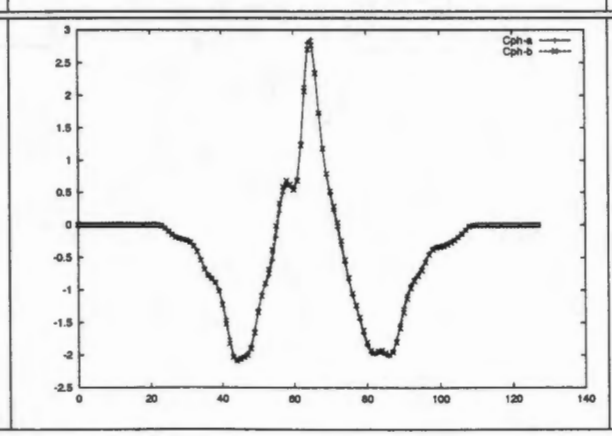
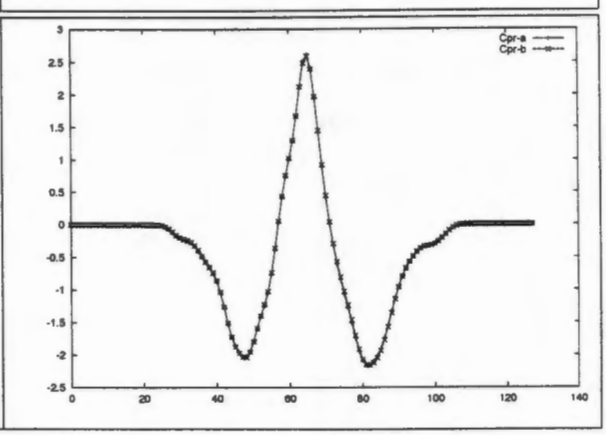
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.928	0.927	0.921	0.877	0.912	0.933	0.897	0.867	0.903	0.899	0.860	0.878	0.895	0.916	0.874	0.932	0.885	0.899	0.926	0.789	0.930	0.896	0.885
2		1.000	0.928	0.962	0.928	0.946	0.969	0.954	0.960	0.937	0.958	0.928	0.922	0.840	0.869	0.915	0.962	0.921	0.834	0.906	0.802	0.860	0.974	0.924
3			1.000	0.969	0.949	0.961	0.945	0.935	0.906	0.957	0.855	0.876	0.881	0.918	0.936	0.907	0.907	0.940	0.913	0.889	0.834	0.896	0.927	0.893
4				1.000	0.976	0.969	0.962	0.960	0.946	0.967	0.915	0.946	0.931	0.903	0.929	0.943	0.944	0.949	0.905	0.916	0.865	0.905	0.972	0.935
5					1.000	0.962	0.948	0.958	0.950	0.955	0.854	0.970	0.910	0.888	0.934	0.959	0.887	0.936	0.906	0.905	0.900	0.898	0.972	0.910
6						1.000	0.957	0.986	0.924	0.968	0.874	0.939	0.964	0.942	0.952	0.976	0.945	0.986	0.926	0.878	0.927	0.916	0.957	0.970
7							1.000	0.966	0.978	0.935	0.943	0.935	0.917	0.853	0.885	0.930	0.953	0.930	0.850	0.910	0.825	0.868	0.974	0.922
8								1.000	0.949	0.957	0.895	0.954	0.973	0.913	0.930	0.985	0.952	0.986	0.894	0.882	0.926	0.892	0.977	0.977
9									1.000	0.921	0.938	0.953	0.887	0.797	0.837	0.912	0.923	0.896	0.804	0.904	0.799	0.825	0.978	0.891
10										1.000	0.894	0.921	0.909	0.943	0.953	0.946	0.938	0.958	0.944	0.941	0.884	0.934	0.942	0.925
11											1.000	0.881	0.868	0.778	0.790	0.844	0.967	0.850	0.775	0.909	0.710	0.822	0.920	0.878
12												1.000	0.938	0.856	0.896	0.961	0.904	0.915	0.874	0.901	0.906	0.893	0.973	0.932
13													1.000	0.909	0.910	0.971	0.949	0.970	0.881	0.828	0.938	0.887	0.939	0.997
14														1.000	0.983	0.927	0.891	0.949	0.989	0.874	0.931	0.969	0.853	0.926
15															1.000	0.952	0.882	0.952	0.987	0.908	0.939	0.973	0.898	0.919
16																1.000	0.918	0.981	0.918	0.875	0.969	0.911	0.958	0.972
17																	1.000	0.939	0.869	0.903	0.831	0.895	0.933	0.962
18																		1.000	0.919	0.855	0.944	0.901	0.941	0.979
19																			1.000	0.909	0.919	0.988	0.855	0.896
20																				1.000	0.786	0.942	0.908	0.843
21																					1.000	0.894	0.868	0.938
22																						1.000	0.868	0.900
23																							1.000	0.935
24																								1.000

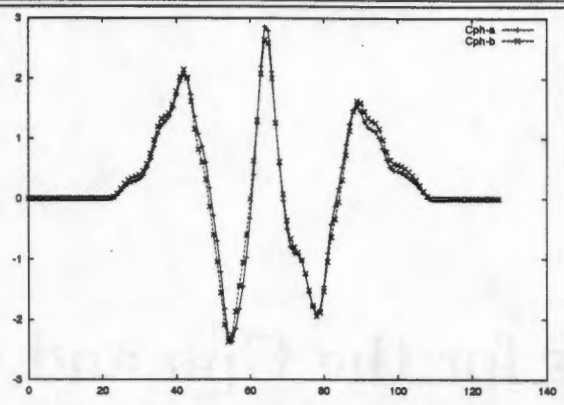
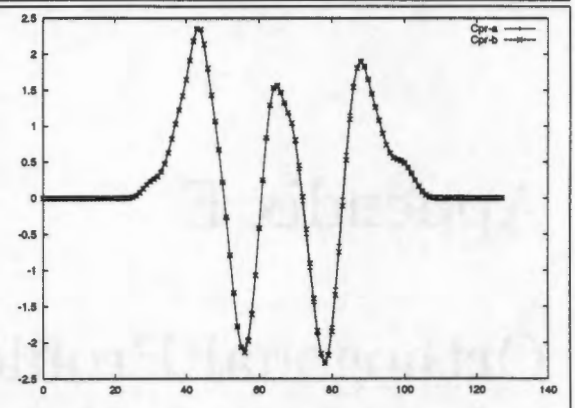
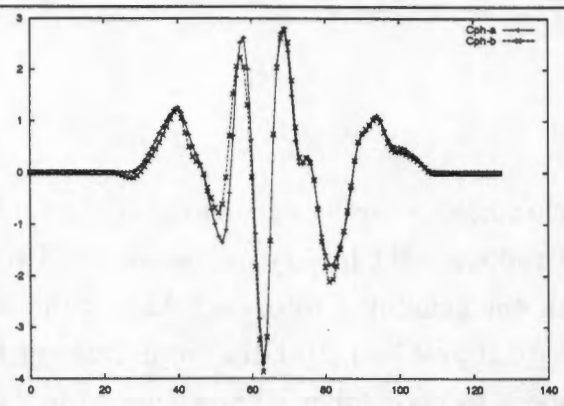
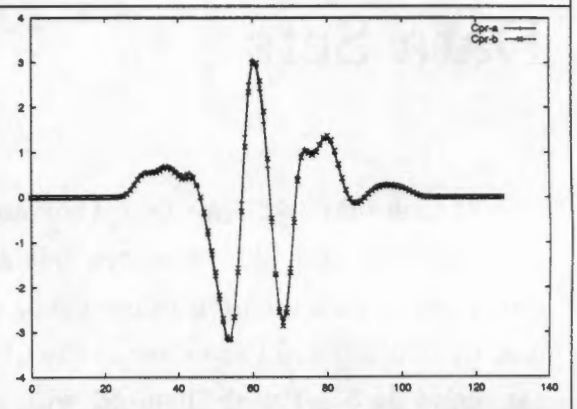
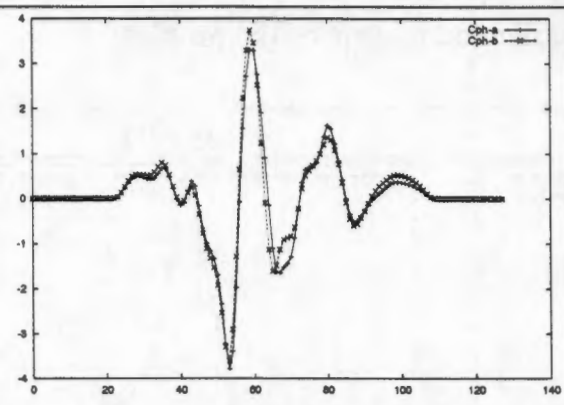
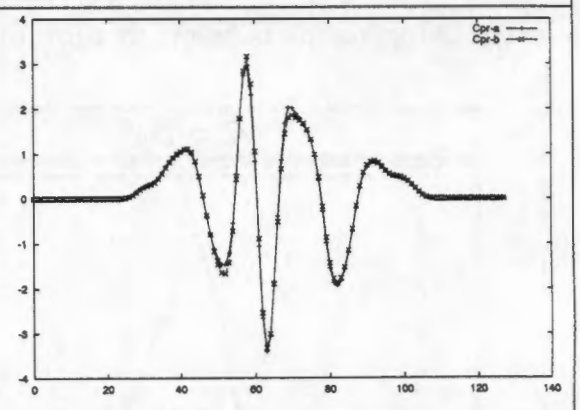
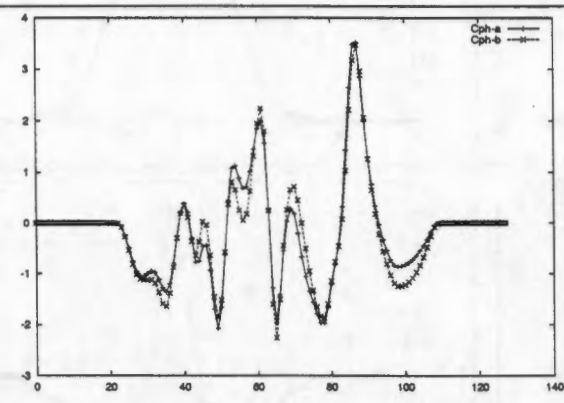
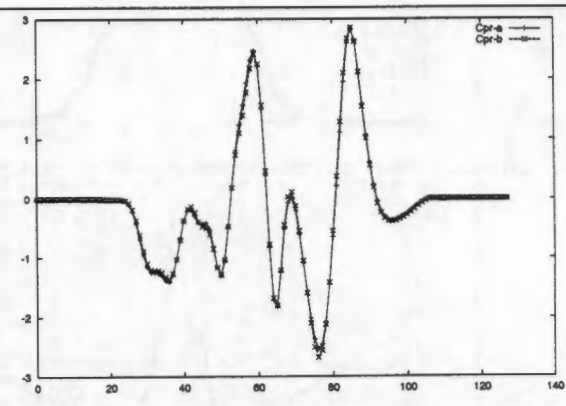
Table D.4: CPR-La library correlation coefficients

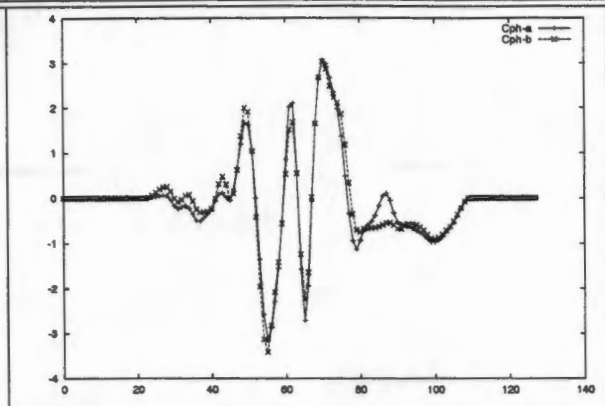
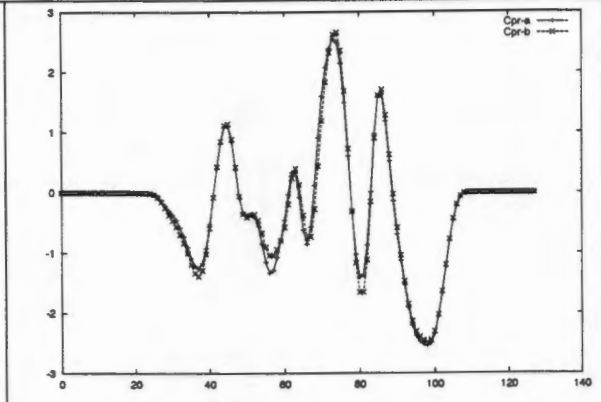
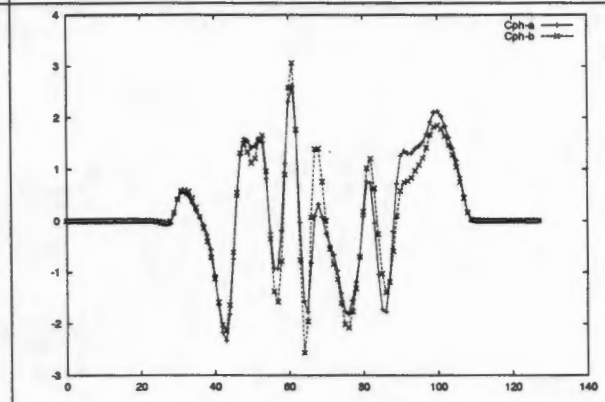
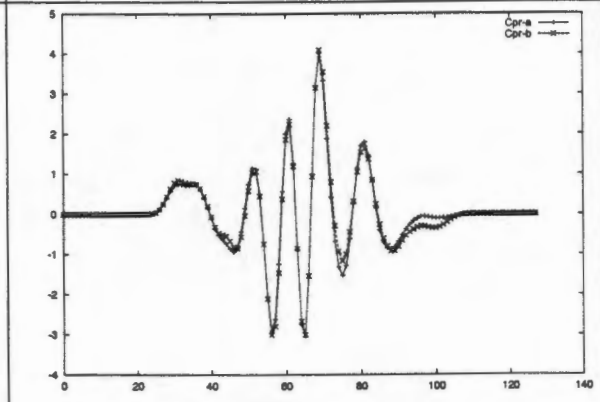
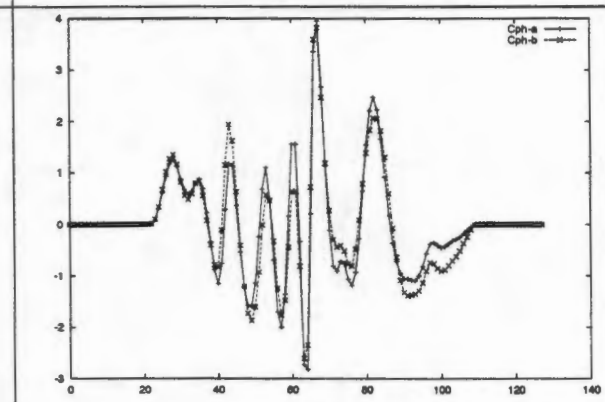
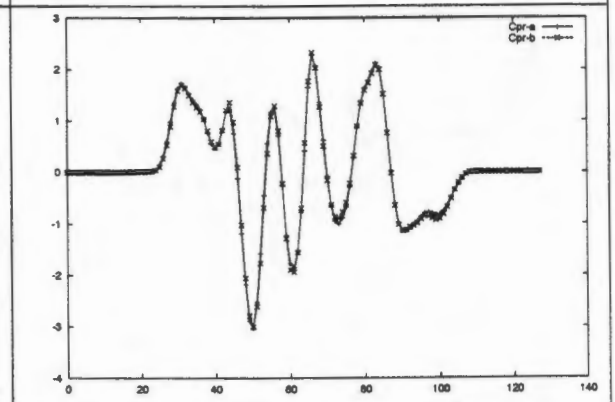
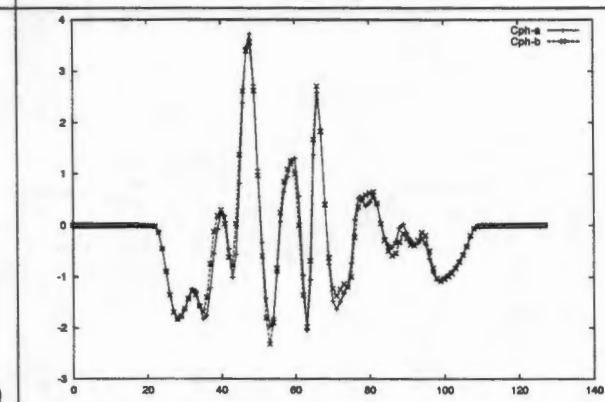
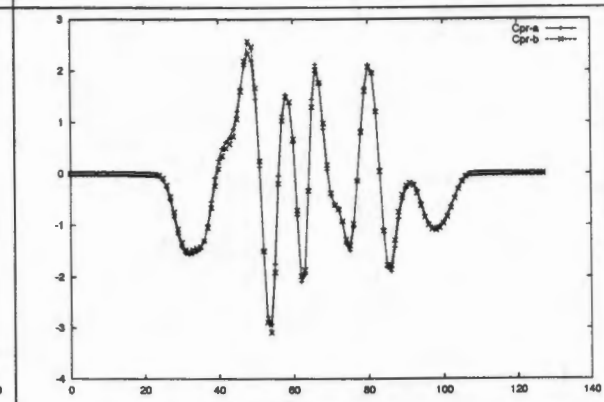
Appendix E

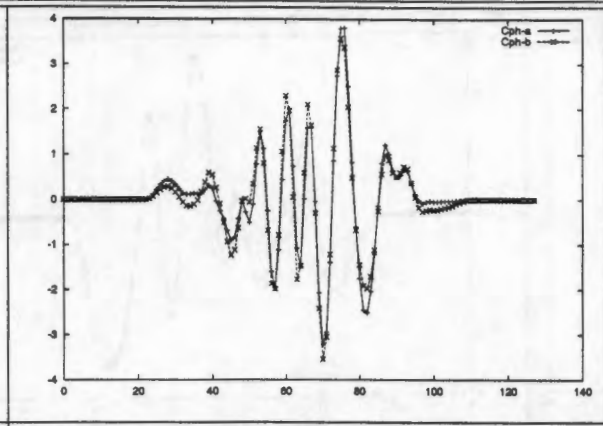
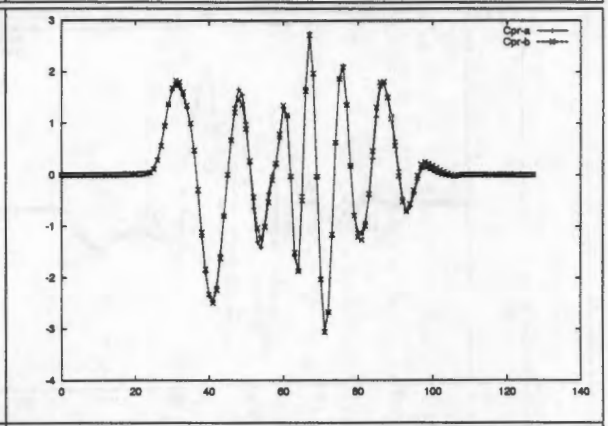
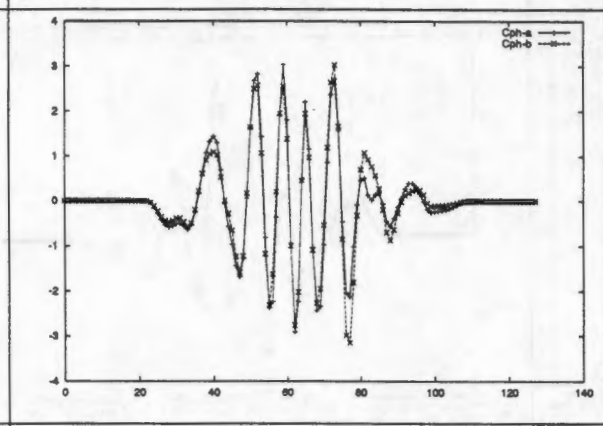
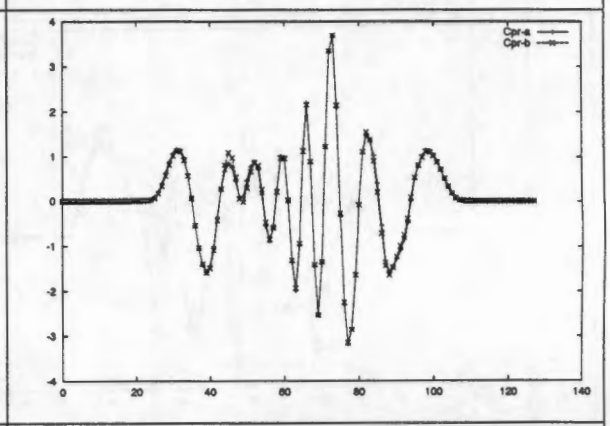
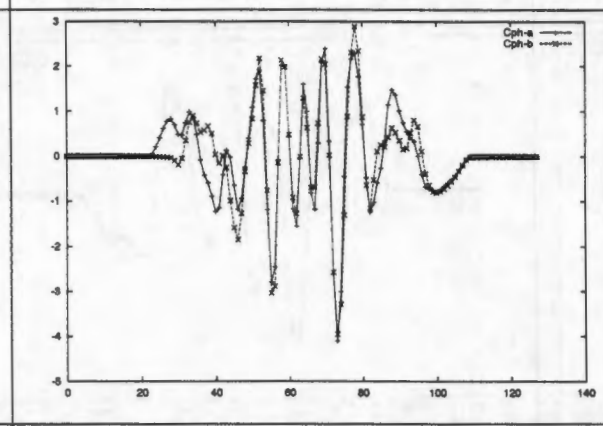
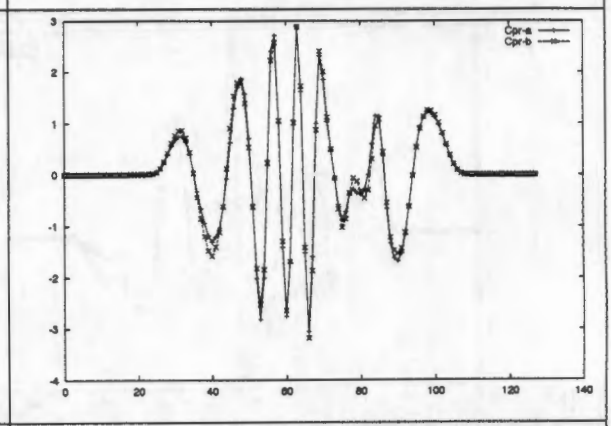
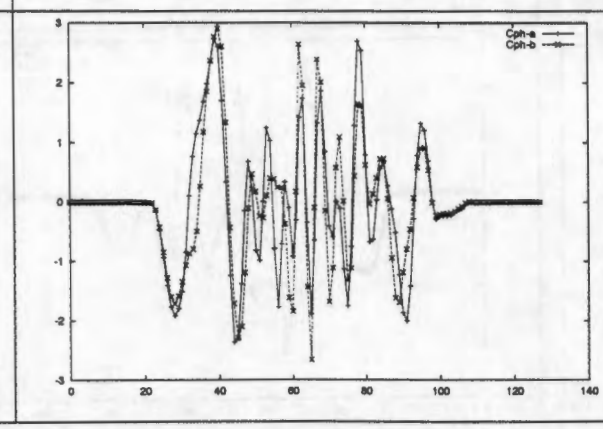
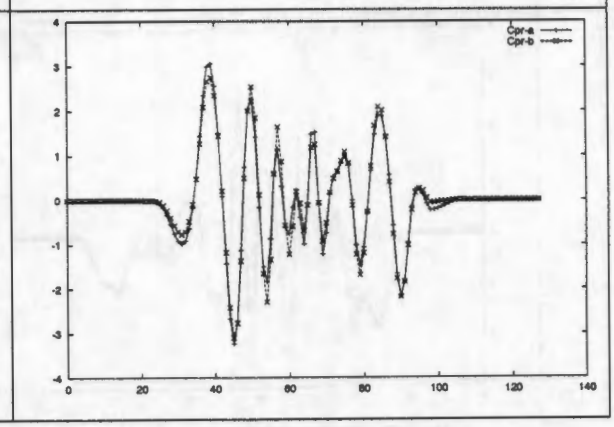
Orthogonal Profiles for the Cph and Cpr Data Sets

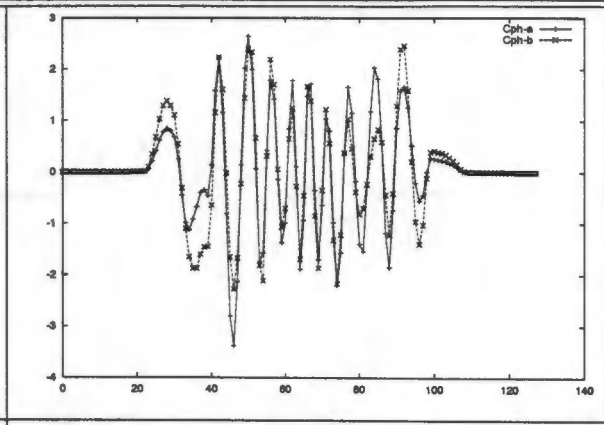
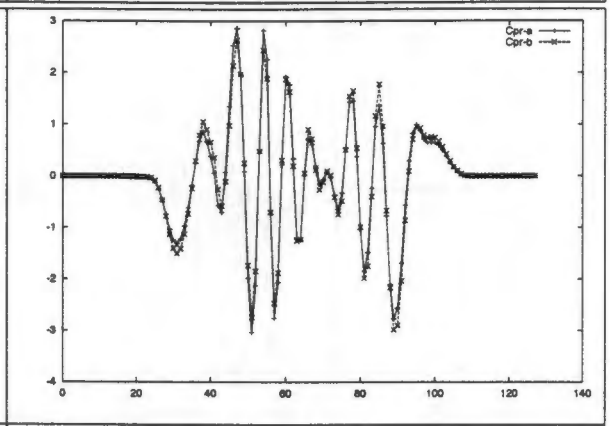
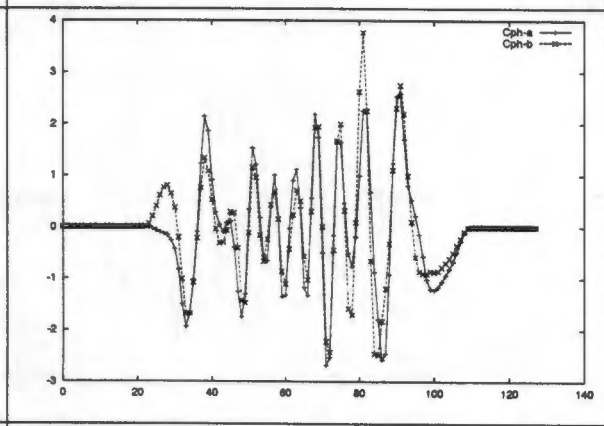
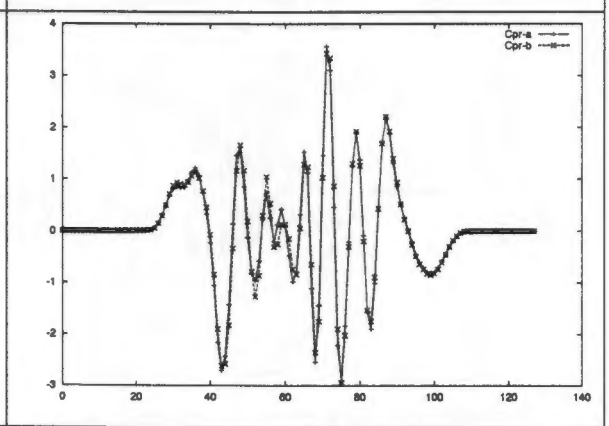
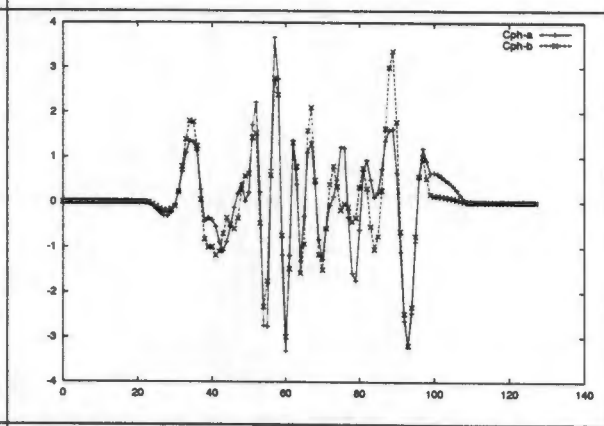
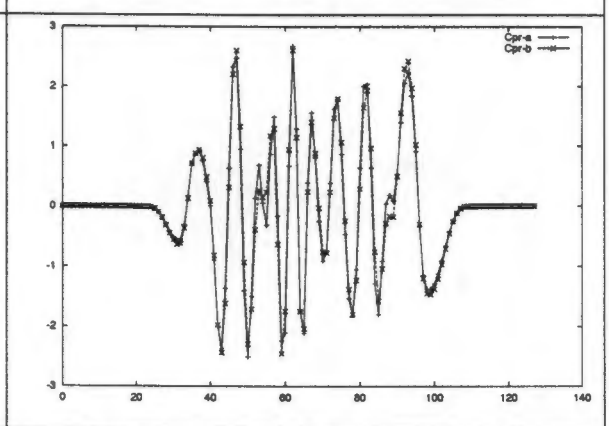
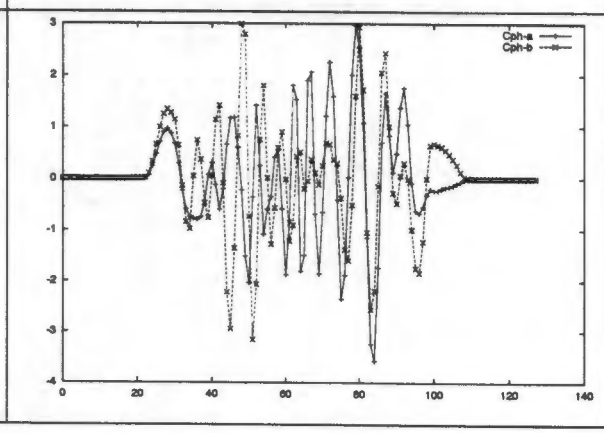
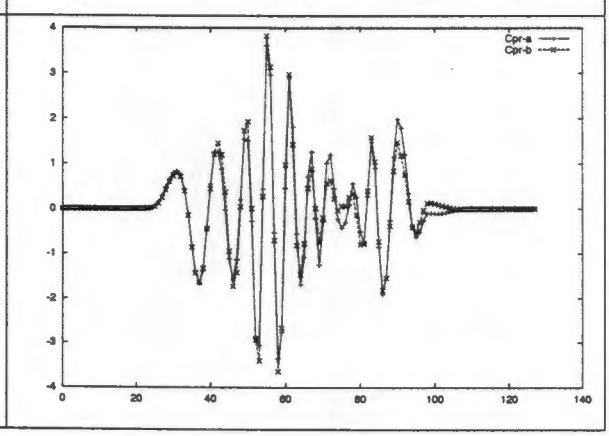
The 24 Cph-OM and Cpr-OM orthogonal chromosome profiles are shown in the table below, with the Cph-OM library on the left and the Cpr-OM library on the right. Two profiles are shown for each chromosome class, with one generated from each half of the library (a and b). Note that in some cases the orthogonal profiles calculated from different halves of the same data set were inverted with respect to each other. These inversions have been corrected in the plots below to allow for easier comparison of the profiles.

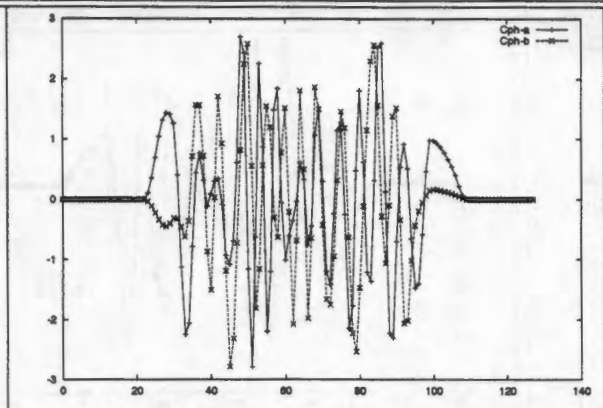
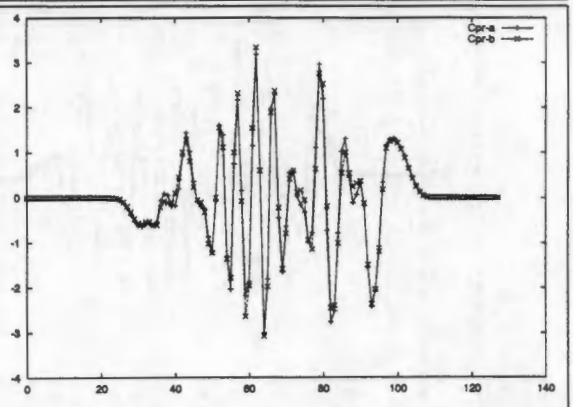
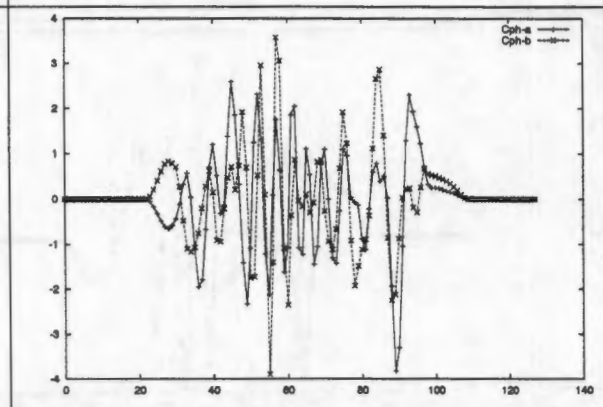
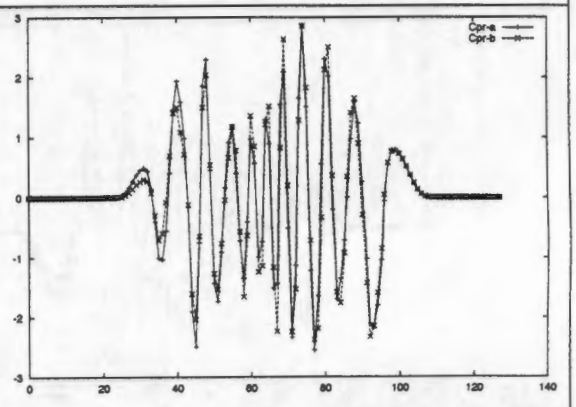
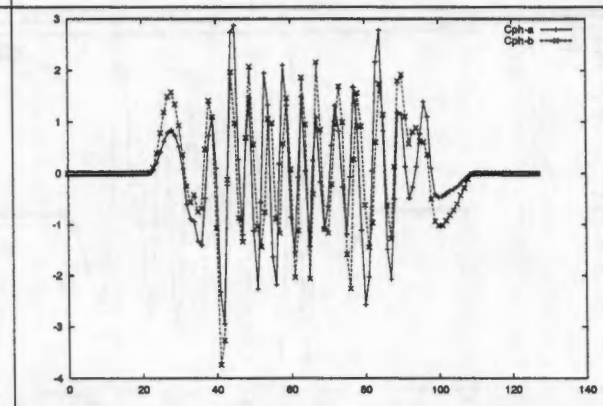
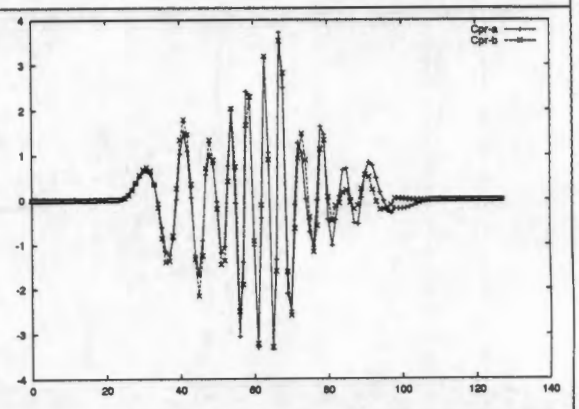
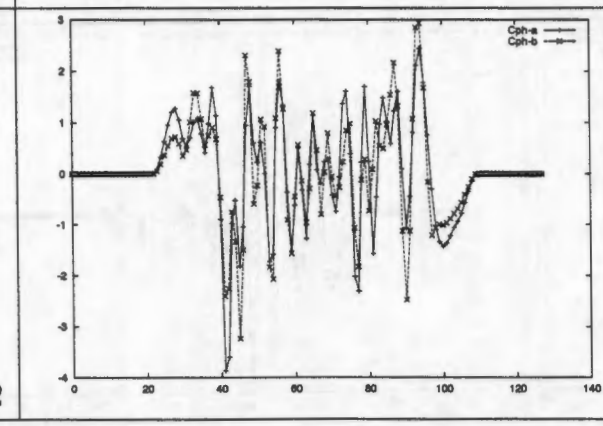
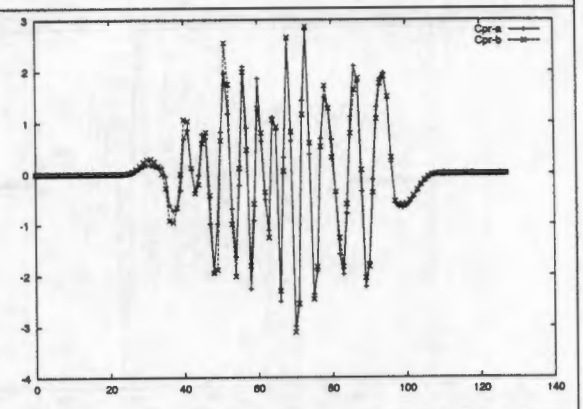
#	Cph-OM	Cpr-OM
1		
2		

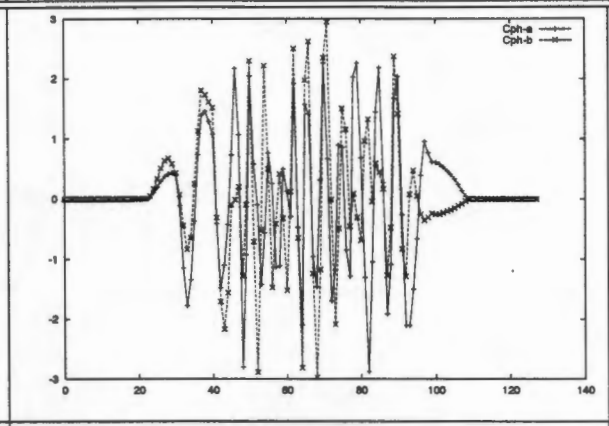
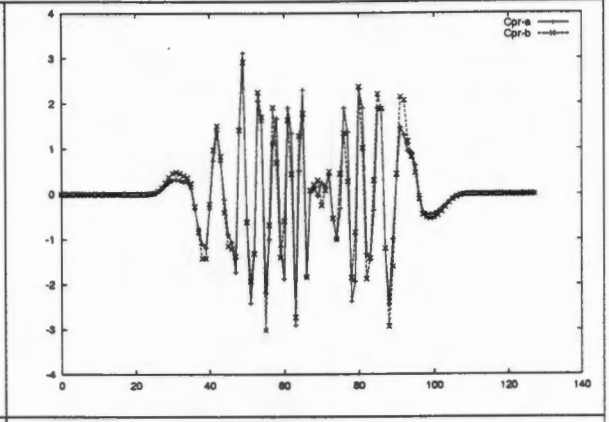
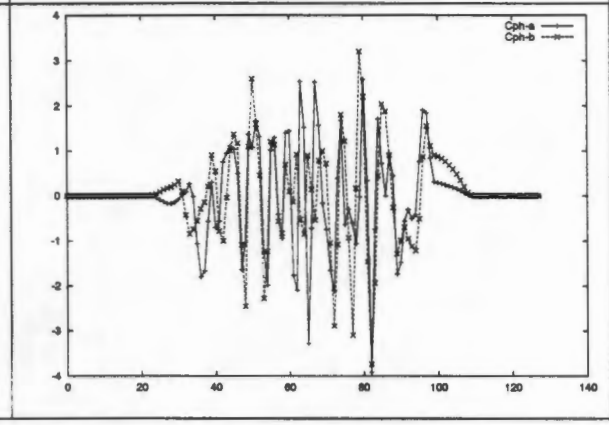
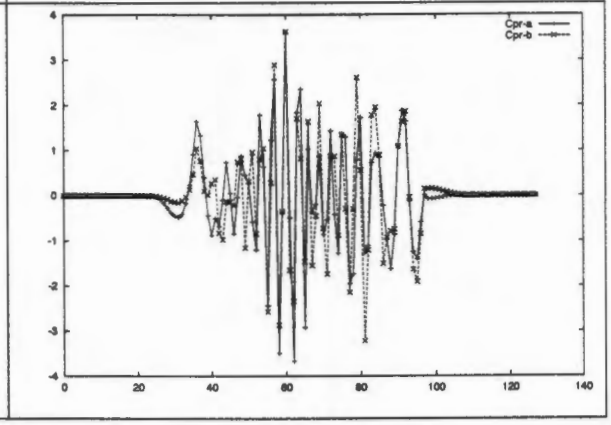
#	Cph-OM	Cpr-OM
3		
4		
5		
6		

#	Cph-OM	Cpr-OM
7		
8		
9		
10		

#	Cph-OM	Cpr-OM
11		
12		
13		
14		

#	Cph-OM	Cpr-OM
15		
16		
17		
18		

#	Cph-OM	Cpr-OM
19		
20		
21		
22		

#	Cph-OM	Cpr-OM
23		
24		

Appendix F

Orthogonal Library Correlation Coefficients

In most situations in which a function is represented in terms of a weighted sum of orthogonal functions, these orthogonal functions are generated algebraically using a theoretical description of the functions being represented. Due to the absence of a theoretical description of chromosome integrated density profiles, the unusual technique of constructing a series of orthogonal functions based on averages of experimental profiles was used. Because of this, a test of the quality of these empirically determined orthogonal profiles was performed.

Two sets of average profiles were calculated for each data set, one from each section of the data set (see Appendix C). A set of orthogonal profiles was constructed from each set of average profiles, leading to two sets of orthogonal profiles calculated for each data set (see Appendix E). In order to test the quality of these empirically determined orthogonal profiles, overlaps (zero-shift correlation coefficients) of orthogonal profiles constructed using the subset a averages of a data set with orthogonal profiles constructed using subset b of the same data set were calculated. These overlaps are shown in the four tables in this appendix, and were calculated for orthogonal profiles in the Cph-OMa and Cph-OMb libraries (Table F.1), the Cph-OLa and Cph-OLb libraries (Table F.2), the Cpr-OMa and Cpr-OMb libraries (Table F.3) and the Cpr-OLa and Cpr-OLb libraries (Table F.4). Numbers on the left of each matrix refer to orthogonal chromosomes calculated from subset a, and numbers along the top refer to chromosomes calculated from subset b.

The full matrices are shown as they are not symmetrical. This is because two sets of profiles are being compared with each other, in contrast to the matrices shown in Appendix D, where one set of profiles is being compared to itself.

If the orthogonal profiles were perfectly determined, then correlating two sets of orthogonal profiles constructed using different subsets of the same data set should lead to a matrix with ones on the diagonal and zeros everywhere else. A brief glance at the tables

in this appendix will convince the reader that these orthogonal profiles do not satisfy this requirement completely, but that it is sufficiently satisfied, especially for the lower numbered orthogonal profiles. Even though the negative correlation coefficients on the diagonal might cause some initial alarm, they are acceptable as they are due to an orthogonal chromosome profile calculated from one section of the data set being inverted with respect to the orthogonal chromosome profile calculated from the other section of the data set.

Examination of Table F.1 shows that for the first 12 orthogonal profiles in the Cph data set, the orthogonality criterion is met rather well, except in a few cases such as the overlaps between orthogonal profiles 6 and 7. A similar situation is visible in Table F.2, except that in a few situations, orthogonal profiles are numbered differently in the two sets so that not all the correlation coefficients close to 1 fall on the diagonal, for example, profiles 7 and 8. Due to the larger size of the Cpr data set, the orthogonal profiles determined from this data set have better inter-subset orthogonality properties than the Cph set, as is visible in Tables F.3 and F.4. Further insight into the similarities of the orthogonal profiles calculated using different subsets can be obtained by examining the plots of the orthogonal profiles in Appendix E.

The choice to use only 12 orthogonal profiles when calculating the γ coefficients using orthogonal chromosomes in Chapter 7 was made based on these matrices and the plots of the orthogonal profiles. Even though more orthogonal profiles could possibly be used for the Cpr data set, it was decided to use the same number for all data sets.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.007	0.002	-0.006	0.006	0.001	-0.002	-0.003	0.000	0.000	0.000	-0.002	0.001	0.001	-0.001	0.000	-0.000	0.000	-0.002	0.000	-0.000	0.000	-0.000	0.001
2	-0.007	0.999	0.033	-0.012	-0.000	-0.004	0.007	0.005	-0.008	0.006	0.006	0.002	-0.009	-0.003	0.005	-0.002	0.001	-0.002	0.002	-0.002	0.003	-0.002	-0.002	-0.001
3	-0.001	-0.034	0.991	-0.075	-0.090	-0.010	0.012	-0.008	0.022	0.035	-0.004	0.007	-0.001	-0.005	-0.010	0.003	0.004	-0.006	0.004	0.013	-0.003	0.003	-0.005	0.007
4	-0.004	-0.008	-0.089	-0.973	-0.197	0.005	-0.036	-0.008	-0.027	-0.044	0.009	0.004	0.018	0.009	-0.002	0.017	-0.014	0.005	-0.003	-0.013	-0.012	0.002	0.015	-0.008
5	-0.007	-0.005	0.072	-0.205	0.972	-0.012	0.052	-0.046	-0.036	0.010	0.010	0.017	0.002	-0.007	-0.011	-0.016	-0.001	-0.001	-0.018	-0.008	-0.005	-0.014	-0.001	0.002
6	-0.001	0.004	0.017	0.007	0.028	0.973	-0.191	0.032	-0.024	-0.059	0.001	-0.003	-0.019	0.019	0.029	0.033	0.006	0.002	0.058	0.034	0.017	0.034	-0.009	0.019
7	0.003	-0.005	-0.015	-0.022	-0.049	0.190	0.975	0.069	0.002	-0.033	-0.011	0.026	0.044	0.007	-0.016	0.015	-0.007	0.014	-0.009	0.004	0.003	0.007	-0.005	-0.007
8	0.003	-0.008	0.020	-0.005	0.032	-0.047	-0.056	0.951	-0.274	-0.052	0.006	0.030	-0.015	-0.056	0.015	-0.041	0.023	0.010	-0.024	-0.036	-0.022	-0.009	0.003	-0.018
9	0.000	0.006	-0.018	-0.036	0.041	0.010	-0.020	0.269	0.955	-0.009	0.048	0.047	-0.039	-0.001	-0.009	-0.046	0.024	-0.028	0.016	0.005	-0.010	0.004	-0.007	0.011
10	0.000	-0.006	-0.036	-0.040	-0.012	0.057	0.019	0.057	-0.010	0.986	0.087	-0.029	-0.017	-0.003	0.047	0.010	0.018	0.013	0.025	0.005	0.048	0.003	-0.013	-0.019
11	-0.001	-0.005	0.008	0.014	-0.007	-0.004	0.013	-0.012	-0.033	-0.087	0.974	-0.187	0.020	-0.007	-0.023	0.042	0.011	-0.009	-0.035	0.021	-0.011	-0.006	-0.007	0.023
12	0.002	-0.004	-0.006	0.011	-0.020	0.004	-0.017	-0.046	-0.044	0.014	0.186	0.969	-0.114	0.011	-0.005	0.030	-0.035	0.009	-0.030	-0.049	0.009	-0.039	0.013	-0.009
13	0.001	-0.010	-0.003	-0.020	-0.009	-0.000	0.051	-0.045	-0.023	-0.022	-0.004	-0.100	-0.911	-0.359	0.087	-0.061	0.080	0.035	-0.018	-0.028	-0.032	-0.001	-0.024	0.026
14	-0.001	0.002	0.002	0.003	0.010	-0.020	0.001	0.053	-0.012	0.005	-0.015	-0.060	-0.366	0.817	-0.350	0.222	0.002	0.077	-0.061	-0.018	0.021	0.032	0.008	-0.051
15	0.001	-0.004	0.012	0.001	0.018	-0.048	0.023	0.012	0.007	-0.043	0.019	-0.005	-0.062	0.327	0.901	0.162	-0.016	-0.057	0.146	0.035	-0.065	0.043	0.084	-0.035
16	0.000	-0.003	0.001	-0.010	-0.018	0.026	0.017	-0.032	-0.047	0.007	0.044	0.014	-0.060	0.265	0.014	-0.891	-0.021	-0.282	0.119	0.062	-0.079	-0.061	0.007	-0.075
17	0.000	-0.001	-0.011	-0.018	0.003	-0.029	0.006	-0.006	-0.027	-0.026	-0.006	0.054	0.039	0.018	-0.012	0.018	0.887	-0.114	0.091	0.323	0.233	0.041	0.026	0.004
18	0.002	-0.002	-0.007	-0.004	0.020	-0.062	0.017	0.037	-0.023	-0.021	0.023	0.017	-0.060	-0.070	-0.180	0.184	-0.232	-0.298	0.823	0.167	0.147	0.055	0.079	0.002
19	-0.000	0.003	-0.006	-0.008	0.003	-0.031	-0.006	0.022	-0.001	-0.003	0.002	0.031	-0.013	0.022	0.019	-0.144	-0.177	0.632	0.094	0.684	0.022	-0.013	-0.048	0.185
20	-0.000	-0.001	-0.007	-0.010	0.000	-0.008	-0.000	0.029	-0.007	-0.002	-0.023	0.008	-0.050	-0.001	0.047	0.072	-0.279	-0.494	-0.466	0.427	0.417	0.155	-0.053	0.160
21	0.000	0.003	-0.009	0.006	-0.002	0.023	-0.002	-0.001	-0.002	0.036	-0.020	0.007	-0.007	-0.045	-0.079	0.189	0.013	-0.263	-0.117	0.415	-0.692	-0.323	0.107	-0.238
22	0.000	-0.003	0.003	-0.004	-0.006	0.018	0.001	0.009	0.012	-0.025	-0.005	-0.047	-0.013	0.024	0.068	0.013	-0.065	0.039	0.009	-0.042	0.401	-0.850	-0.026	-0.150
23	-0.001	0.000	-0.002	0.003	-0.006	0.010	0.006	0.011	-0.001	0.025	-0.021	-0.014	-0.023	0.005	-0.040	0.024	0.002	-0.139	-0.010	-0.055	-0.056	-0.180	0.515	0.653
24	-0.001	-0.000	-0.010	-0.011	-0.000	-0.021	0.008	0.009	-0.015	0.005	-0.009	0.016	0.017	0.073	0.034	-0.003	0.049	-0.059	0.077	-0.056	-0.225	-0.112	-0.569	0.506

Table F.1: CPH-OMa and CPH-OMb orthogonal library correlation coefficients

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.002	-0.006	0.005	0.001	-0.002	0.000	-0.005	0.000	0.000	-0.002	0.004	-0.002	-0.001	0.002	-0.003	-0.001	-0.002	-0.001	-0.000	-0.001	0.000	-0.000	-0.001
2	-0.002	0.995	-0.080	-0.047	0.008	-0.005	-0.009	0.020	0.009	0.001	0.021	-0.005	0.012	-0.003	-0.002	0.012	0.006	0.007	0.007	-0.005	-0.002	-0.002	0.001	-0.005
3	-0.006	-0.083	-0.994	-0.059	-0.014	-0.019	0.016	-0.013	-0.002	0.003	0.012	0.001	-0.001	-0.005	0.017	0.005	-0.005	0.012	0.007	-0.001	0.000	-0.006	0.002	0.003
4	0.006	-0.042	0.058	-0.981	0.044	0.164	-0.019	0.021	-0.018	0.025	0.010	0.007	-0.009	-0.001	-0.003	-0.009	0.016	-0.013	0.005	-0.007	-0.003	0.005	0.001	-0.000
5	-0.001	-0.004	-0.020	0.076	0.972	0.199	0.013	-0.039	-0.072	0.003	0.031	-0.027	0.005	0.019	0.011	0.005	-0.008	-0.005	0.011	0.002	0.005	-0.000	0.002	0.002
6	0.002	0.009	-0.025	0.149	-0.204	0.960	-0.047	0.033	0.048	-0.068	0.018	-0.010	-0.006	-0.004	0.025	-0.020	-0.002	0.026	0.021	-0.008	0.014	0.003	0.011	0.007
7	0.005	-0.022	-0.013	0.018	0.047	-0.027	0.030	0.994	0.069	0.024	0.005	-0.015	-0.011	0.022	-0.006	0.011	0.014	-0.005	0.004	0.003	-0.002	0.018	-0.015	-0.003
8	-0.000	0.010	0.015	-0.013	-0.018	0.049	0.992	-0.034	0.072	0.034	0.003	0.003	0.005	-0.021	-0.009	0.048	-0.012	-0.009	-0.014	-0.006	-0.016	0.006	-0.013	0.005
9	-0.001	-0.009	-0.000	-0.008	0.071	-0.013	-0.078	-0.073	0.947	0.289	-0.001	0.003	-0.004	-0.025	-0.020	-0.024	0.016	0.026	0.004	0.015	0.006	-0.010	0.000	0.007
10	0.000	-0.004	0.001	-0.041	0.039	-0.066	0.013	-0.001	0.289	-0.951	0.010	-0.008	0.022	0.013	0.015	0.036	-0.026	0.007	-0.008	0.007	0.011	-0.016	-0.009	-0.015
11	-0.004	0.001	0.002	0.013	0.015	0.005	-0.000	0.011	0.003	-0.005	0.272	0.935	0.070	0.020	0.089	-0.110	-0.015	-0.065	-0.009	0.017	-0.044	0.085	-0.033	-0.053
12	-0.002	0.017	-0.013	-0.005	0.032	0.025	0.006	0.012	-0.004	-0.011	-0.935	0.269	0.177	0.016	-0.061	-0.038	-0.001	0.077	0.009	0.046	0.018	-0.015	0.021	0.018
13	0.002	-0.017	0.003	-0.003	-0.005	0.007	-0.019	0.010	-0.006	0.016	0.119	-0.091	0.888	-0.323	0.074	0.142	0.202	-0.020	-0.057	0.010	-0.080	0.020	-0.035	-0.006
14	-0.001	0.002	-0.001	0.007	0.026	0.000	-0.010	0.016	-0.025	-0.039	-0.033	0.065	-0.320	-0.890	-0.223	-0.083	0.146	0.084	-0.048	0.005	-0.034	0.046	0.004	0.016
15	-0.003	0.008	0.012	-0.002	0.001	-0.018	0.005	0.001	0.017	0.010	-0.129	-0.063	-0.156	-0.176	0.929	-0.026	0.085	-0.179	0.010	0.034	0.001	0.096	-0.022	-0.042
16	0.001	-0.005	-0.004	0.010	0.001	0.019	-0.023	-0.021	0.005	-0.004	-0.026	0.105	-0.195	0.162	-0.055	0.601	0.699	-0.031	-0.159	0.032	0.029	0.077	-0.001	0.109
17	-0.002	0.009	-0.016	0.014	0.002	0.001	0.036	-0.003	0.002	-0.041	-0.012	-0.076	0.029	0.122	-0.110	-0.696	0.491	-0.409	-0.181	-0.025	0.031	-0.065	0.030	0.077
18	0.001	-0.003	0.011	-0.011	0.006	-0.023	0.029	0.000	-0.024	-0.011	0.071	-0.025	0.009	0.120	0.153	-0.307	0.240	0.855	-0.108	0.145	-0.050	0.040	-0.019	0.008
19	0.001	-0.008	0.009	0.004	-0.004	-0.023	0.022	0.001	-0.008	-0.016	0.033	0.031	0.004	-0.011	0.005	-0.040	0.251	0.036	0.912	0.050	-0.087	-0.202	0.051	0.095
20	-0.000	-0.006	0.005	0.005	0.009	-0.009	-0.001	0.008	0.007	0.003	-0.023	0.048	-0.007	-0.002	0.080	0.015	0.093	0.157	-0.070	-0.886	-0.101	-0.268	0.004	-0.075
21	0.000	0.000	0.005	-0.002	0.001	0.012	0.001	0.018	-0.016	0.017	0.035	0.060	-0.037	-0.048	0.070	0.073	0.009	0.009	-0.240	0.367	-0.127	-0.799	-0.043	-0.069
22	0.001	-0.002	0.003	-0.008	-0.001	-0.002	0.005	0.006	-0.006	0.012	0.026	0.058	0.029	-0.064	0.022	0.024	-0.051	0.021	0.049	-0.100	0.776	-0.252	-0.031	-0.002
23	-0.000	0.003	-0.001	0.004	-0.001	-0.004	-0.001	-0.010	-0.011	0.009	-0.007	-0.038	0.015	0.003	-0.013	-0.023	0.088	0.012	0.040	0.037	0.293	0.014	-0.683	0.029
24	0.000	0.003	0.001	0.004	0.000	-0.002	-0.003	0.002	-0.004	0.008	-0.016	-0.032	0.002	0.003	0.011	0.034	0.037	-0.039	0.006	0.029	-0.014	-0.103	0.419	-0.243

Table F.2: CPH-OLa and CPH-OLb orthogonal library correlation coefficients

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.003	0.000	-0.002	0.002	-0.003	0.001	0.001	0.001	-0.000	-0.001	0.000	0.000	0.000	0.000	0.000	-0.000	-0.000	-0.000	0.000	-0.000	-0.000	0.001	-0.000
2	-0.003	1.000	0.007	0.000	-0.007	0.006	-0.001	-0.003	-0.000	-0.001	0.001	0.002	0.001	-0.001	-0.001	0.002	0.000	-0.002	-0.001	0.000	0.001	-0.001	-0.000	-0.000
3	-0.000	-0.008	1.000	0.005	-0.008	0.014	-0.015	-0.000	-0.012	-0.003	-0.000	-0.002	-0.000	-0.006	-0.003	-0.000	-0.002	0.000	-0.003	0.000	-0.001	0.000	0.002	0.000
4	0.002	0.000	-0.004	0.998	0.064	0.005	-0.006	0.006	-0.014	-0.002	-0.001	-0.000	-0.004	-0.001	-0.003	-0.001	-0.006	-0.001	0.000	0.001	-0.003	0.002	0.001	-0.001
5	-0.002	0.007	0.008	-0.064	0.997	-0.022	-0.000	0.017	-0.001	0.013	-0.003	0.006	-0.002	-0.007	-0.001	-0.004	0.002	0.004	-0.002	0.000	0.001	-0.000	0.001	0.001
6	0.003	-0.006	-0.014	-0.006	0.022	0.998	0.003	-0.039	0.019	0.003	0.008	-0.009	-0.000	0.004	0.008	0.001	0.003	-0.005	0.003	0.001	0.003	0.001	-0.003	0.001
7	-0.001	0.001	0.015	0.005	-0.001	0.002	0.993	0.108	-0.026	-0.020	-0.008	-0.002	0.011	-0.007	-0.002	0.007	0.006	-0.000	-0.001	-0.006	-0.002	-0.004	-0.001	0.001
8	-0.000	0.002	-0.002	-0.006	-0.016	0.038	-0.107	0.992	0.035	-0.024	0.012	-0.009	0.001	0.003	-0.004	-0.002	-0.002	0.000	0.009	-0.001	-0.007	0.001	0.003	-0.003
9	0.001	0.000	-0.013	-0.015	-0.003	0.020	-0.028	0.032	-0.998	0.032	0.004	-0.002	-0.017	0.001	-0.003	-0.000	-0.014	-0.001	0.002	-0.005	-0.004	0.000	0.003	-0.001
10	-0.000	-0.001	-0.004	-0.004	0.013	0.003	-0.018	-0.024	-0.032	-0.997	-0.046	0.004	-0.017	0.007	0.019	-0.014	-0.003	-0.003	0.006	0.010	0.010	0.005	-0.005	0.001
11	-0.001	0.001	-0.001	-0.001	-0.003	0.008	-0.008	0.013	-0.002	0.047	-0.998	0.013	-0.004	-0.004	0.016	-0.014	-0.014	-0.029	0.003	0.007	-0.007	0.008	-0.007	0.006
12	-0.000	-0.002	0.002	0.000	-0.006	0.010	0.001	0.008	-0.001	0.002	0.013	0.998	0.002	-0.033	-0.009	0.018	-0.031	0.003	0.005	-0.008	-0.006	0.006	0.008	0.001
13	-0.000	-0.001	-0.001	0.003	0.001	0.001	-0.012	-0.001	-0.018	-0.016	-0.006	-0.003	0.994	-0.090	0.002	-0.021	0.029	0.032	-0.009	-0.008	-0.009	-0.011	-0.003	-0.003
14	0.000	-0.001	-0.006	-0.000	-0.006	0.003	-0.007	0.002	0.000	-0.005	0.003	-0.031	-0.091	-0.986	0.095	0.063	0.047	0.031	0.002	-0.032	-0.007	0.000	-0.014	-0.004
15	-0.000	0.001	0.004	0.004	0.001	-0.008	0.002	0.006	-0.004	0.019	0.015	0.015	0.003	0.099	0.989	-0.005	0.074	0.030	-0.050	-0.004	0.005	-0.025	-0.009	-0.005
16	0.000	0.002	-0.000	-0.001	-0.005	0.002	0.007	-0.002	0.001	0.012	0.012	0.015	-0.029	-0.056	-0.002	-0.989	0.012	0.107	0.026	-0.014	-0.052	0.026	-0.002	-0.004
17	-0.000	0.000	-0.002	-0.006	0.001	0.002	0.006	-0.001	0.013	0.001	0.011	-0.032	0.022	-0.039	0.071	-0.004	-0.986	0.099	-0.077	-0.007	0.011	-0.010	0.030	0.030
18	0.000	-0.002	-0.000	-0.002	0.003	-0.005	0.001	0.001	0.001	0.003	0.031	0.001	0.028	-0.039	0.035	-0.108	-0.089	-0.980	-0.060	-0.062	0.065	0.018	-0.019	-0.010
19	0.000	0.001	0.003	-0.001	0.002	-0.004	0.003	-0.008	0.002	0.007	0.007	-0.007	0.014	-0.000	0.058	0.018	-0.076	-0.058	0.988	0.072	-0.003	-0.014	0.012	0.016
20	0.000	0.000	0.000	0.000	0.001	0.001	-0.006	-0.004	0.005	-0.010	-0.006	-0.010	-0.006	0.035	-0.002	0.028	0.004	0.033	0.072	-0.965	-0.214	0.031	0.018	0.041
21	-0.000	0.001	-0.001	-0.003	0.002	0.002	0.000	-0.006	0.003	-0.008	0.010	-0.005	-0.003	-0.001	0.010	0.037	-0.015	-0.081	-0.026	0.216	-0.967	-0.005	0.054	0.036
22	0.000	0.001	0.000	-0.002	0.000	-0.001	0.004	-0.000	-0.000	0.003	0.007	-0.006	0.011	0.005	0.025	0.029	-0.003	0.018	0.007	0.026	0.006	0.980	0.112	0.050
23	0.000	-0.000	0.002	0.001	0.001	-0.003	-0.000	0.004	-0.002	0.005	0.009	0.006	0.001	0.012	-0.004	0.005	-0.019	0.016	0.008	0.006	-0.040	0.095	-0.944	0.264
24	-0.000	-0.000	-0.000	0.001	-0.001	0.001	-0.002	0.003	-0.002	-0.002	-0.001	0.004	-0.004	0.005	-0.000	-0.006	0.031	0.001	-0.013	-0.001	0.036	-0.113	0.229	0.818

Table F.3: CPR-OMa and CPR-OMb orthogonal library correlation coefficients

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1.000	0.006	0.004	0.001	0.004	-0.001	0.001	0.000	-0.000	0.001	-0.000	0.001	-0.000	0.000	-0.000	0.000	0.000	-0.000	-0.000	-0.000	0.000	-0.000	0.001	-0.003
2	-0.007	0.999	0.032	0.008	-0.000	-0.008	0.003	0.004	-0.001	0.003	0.002	0.001	-0.001	0.001	0.000	0.001	-0.000	0.000	-0.000	-0.001	0.001	-0.000	-0.000	0.001
3	-0.004	-0.032	0.999	-0.004	0.011	0.001	-0.008	-0.005	-0.004	-0.005	-0.000	0.001	0.000	-0.001	0.002	0.001	0.002	-0.001	-0.001	0.001	0.000	-0.001	-0.000	-0.000
4	0.001	0.008	-0.004	-0.999	-0.016	0.011	-0.026	0.003	-0.009	-0.003	0.000	-0.003	-0.002	-0.003	0.004	-0.001	0.001	0.001	-0.001	0.000	0.002	-0.000	-0.001	0.000
5	0.004	-0.001	0.011	0.016	-1.000	0.018	-0.000	-0.015	-0.002	0.000	-0.005	-0.002	0.003	-0.004	-0.001	0.001	-0.001	-0.002	0.001	0.000	-0.001	-0.000	0.000	-0.001
6	0.001	0.009	-0.002	0.012	0.019	0.999	-0.027	-0.012	-0.002	-0.016	-0.008	0.000	0.002	0.002	0.005	0.006	0.005	-0.006	-0.003	0.002	-0.001	-0.002	0.002	0.000
7	-0.001	-0.003	0.008	-0.026	-0.000	0.029	0.998	0.032	0.003	0.020	-0.000	0.007	0.006	0.007	-0.002	-0.001	-0.003	0.003	-0.004	-0.001	0.002	0.002	0.001	0.003
8	-0.000	-0.004	0.005	0.004	-0.014	0.011	-0.031	0.999	-0.002	-0.036	-0.006	-0.006	-0.001	0.005	-0.005	-0.004	-0.009	0.000	0.002	-0.003	-0.001	0.003	0.001	0.001
9	0.000	0.001	0.004	-0.009	-0.002	0.003	-0.004	0.004	0.999	0.039	0.005	0.002	-0.014	0.002	0.001	-0.000	0.003	0.000	0.002	0.001	-0.001	-0.003	-0.003	-0.003
10	0.001	0.003	-0.005	0.002	-0.000	-0.016	0.021	-0.036	0.039	-0.997	-0.040	-0.017	-0.014	0.010	-0.006	-0.020	-0.009	0.010	0.002	-0.004	0.001	0.002	-0.000	0.002
11	0.000	-0.002	-0.000	0.000	-0.005	0.007	0.001	0.005	-0.004	-0.042	0.995	0.078	-0.027	-0.025	0.014	-0.009	0.005	0.005	-0.000	0.001	0.010	0.007	-0.002	0.000
12	0.001	0.000	0.001	0.002	0.002	0.001	0.007	-0.005	0.002	0.014	0.080	-0.995	0.048	0.005	-0.001	-0.015	-0.023	0.005	0.007	-0.011	-0.005	0.003	0.002	-0.007
13	-0.000	0.002	-0.000	-0.001	0.003	-0.002	-0.005	0.002	0.015	-0.017	0.017	0.047	0.983	-0.162	0.005	-0.020	0.040	0.009	-0.006	0.008	0.026	0.011	0.009	0.005
14	0.000	-0.001	0.001	-0.003	-0.003	-0.002	-0.008	-0.004	-0.000	0.006	0.027	0.015	0.159	0.984	0.054	-0.031	0.017	0.001	0.010	-0.020	0.015	0.003	0.001	-0.013
15	0.001	0.000	-0.002	0.004	-0.000	-0.005	0.003	0.005	-0.001	-0.007	-0.015	-0.004	-0.013	-0.051	0.996	0.010	0.027	-0.033	-0.012	0.014	-0.012	-0.000	-0.004	-0.021
16	0.000	-0.001	-0.000	-0.001	0.001	-0.005	0.001	0.003	0.001	-0.018	0.011	-0.012	0.024	0.027	-0.003	0.990	-0.014	0.110	0.006	-0.042	-0.006	-0.008	0.036	0.020
17	-0.000	-0.000	-0.001	0.002	-0.002	-0.002	0.001	0.008	-0.003	-0.004	-0.007	-0.022	-0.043	-0.009	-0.016	-0.025	0.934	0.331	-0.082	-0.039	0.041	0.026	0.007	0.027
18	0.001	-0.000	0.002	0.001	-0.001	0.007	-0.004	-0.003	-0.000	0.013	-0.004	0.013	0.006	0.002	0.038	-0.104	-0.325	0.926	-0.039	0.019	-0.079	-0.080	-0.015	-0.023
19	-0.000	0.000	0.001	-0.001	0.001	0.003	0.003	-0.002	-0.002	0.003	-0.000	0.008	0.003	-0.016	0.015	-0.025	0.056	0.054	0.957	-0.207	-0.099	-0.031	0.069	0.070
20	-0.000	0.000	-0.002	0.000	0.001	-0.003	0.003	0.004	-0.002	-0.006	0.003	-0.013	-0.000	0.020	-0.018	0.042	0.078	-0.017	0.195	0.917	-0.059	-0.272	0.013	-0.039
21	-0.000	-0.000	-0.001	-0.000	0.000	-0.003	0.001	0.002	-0.001	-0.003	0.009	0.000	0.011	-0.002	-0.002	-0.018	0.035	-0.077	-0.125	-0.260	-0.185	-0.893	0.232	0.038
22	0.000	0.001	0.000	-0.002	0.001	-0.001	0.001	-0.002	-0.000	-0.002	0.010	0.006	0.020	0.011	-0.013	-0.003	0.041	-0.041	-0.092	0.034	-0.951	0.216	-0.000	0.028
23	0.002	-0.001	0.000	0.001	0.000	0.002	0.002	0.001	-0.004	0.001	-0.001	0.001	-0.000	-0.007	0.007	0.018	0.008	0.020	0.030	-0.020	0.013	-0.201	-0.755	0.110
24	-0.003	-0.000	0.000	-0.001	0.001	0.001	0.002	-0.000	-0.003	-0.002	0.002	0.010	0.009	-0.012	-0.014	0.010	0.019	-0.013	0.026	0.012	-0.010	-0.078	-0.078	-0.810

Table F.4: CPR-OLa and CPR-OLb orthogonal library correlation coefficients

Appendix G

Plots of Chromosome Length Versus Centromeric Index

This appendix contains plots of normalised centromeric index against normalised length for the Copenhagen, Edinburgh and Philadelphia data sets. The shapes of the points indicate the Denver class to which the chromosome belongs.

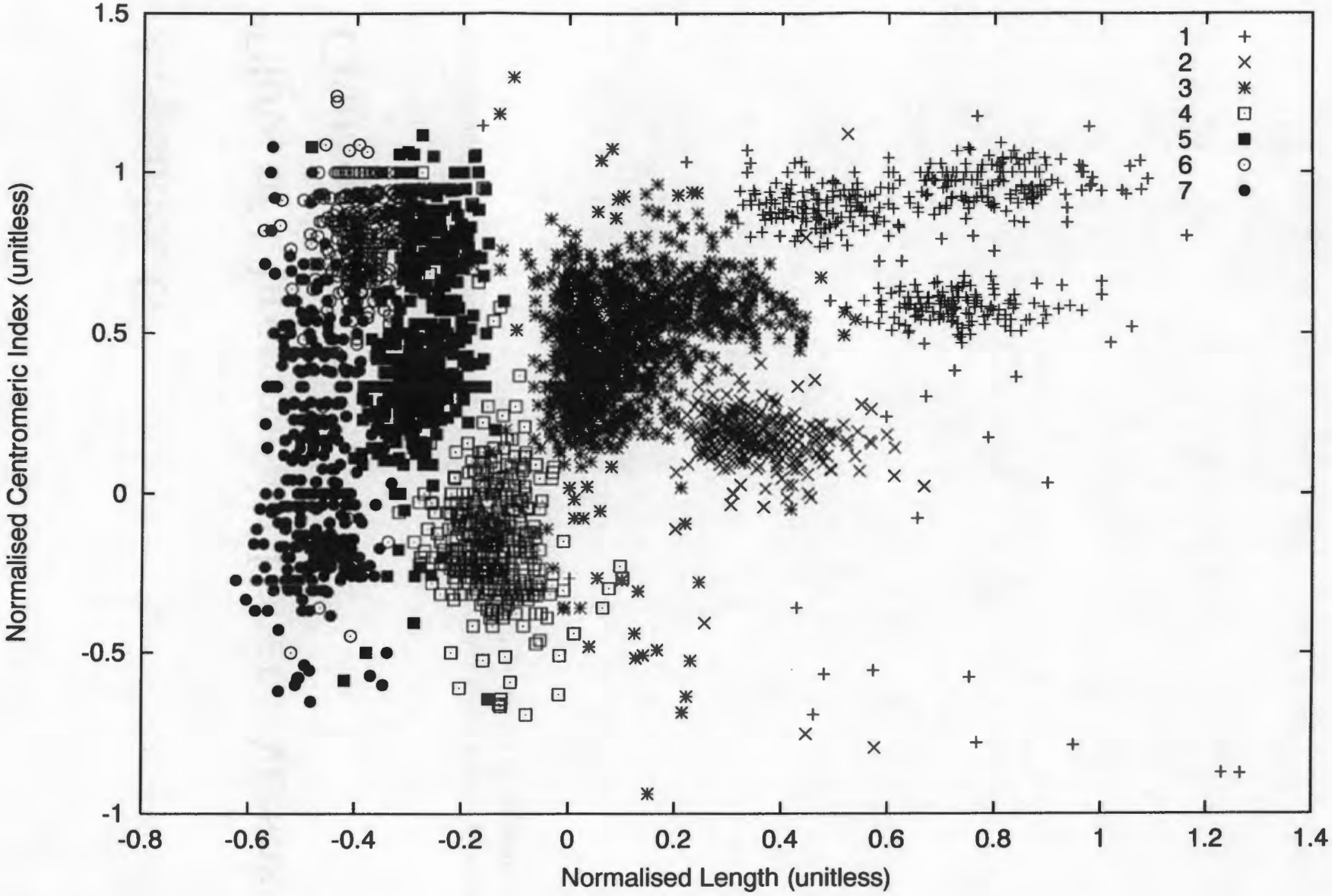


Figure G.1: Plot of normalised centromeric index against normalised length for the first half of the Copenhagen data set.

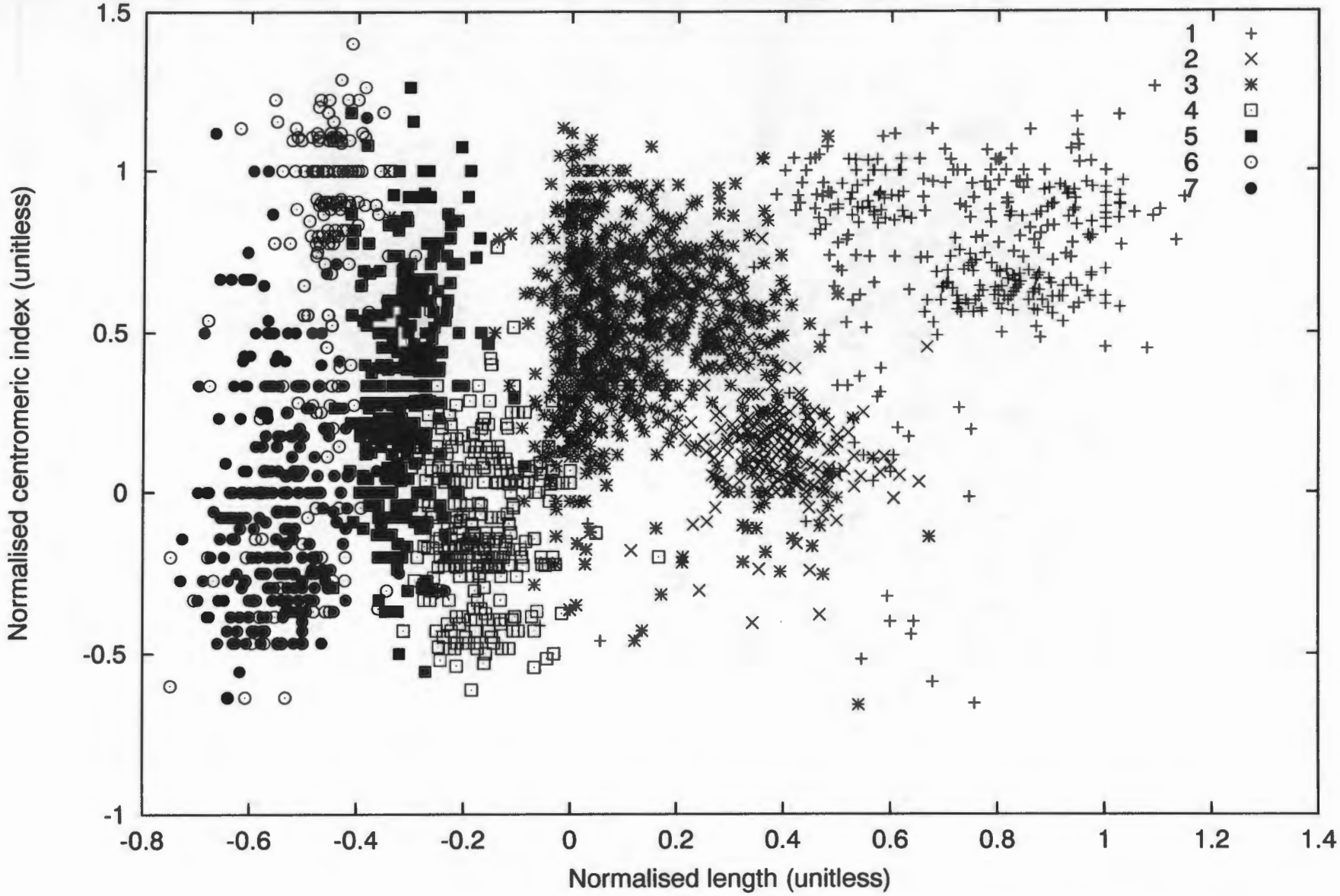


Figure G.2: Plot of normalised centromeric index against normalised length for the first half of the Edinburgh data set.

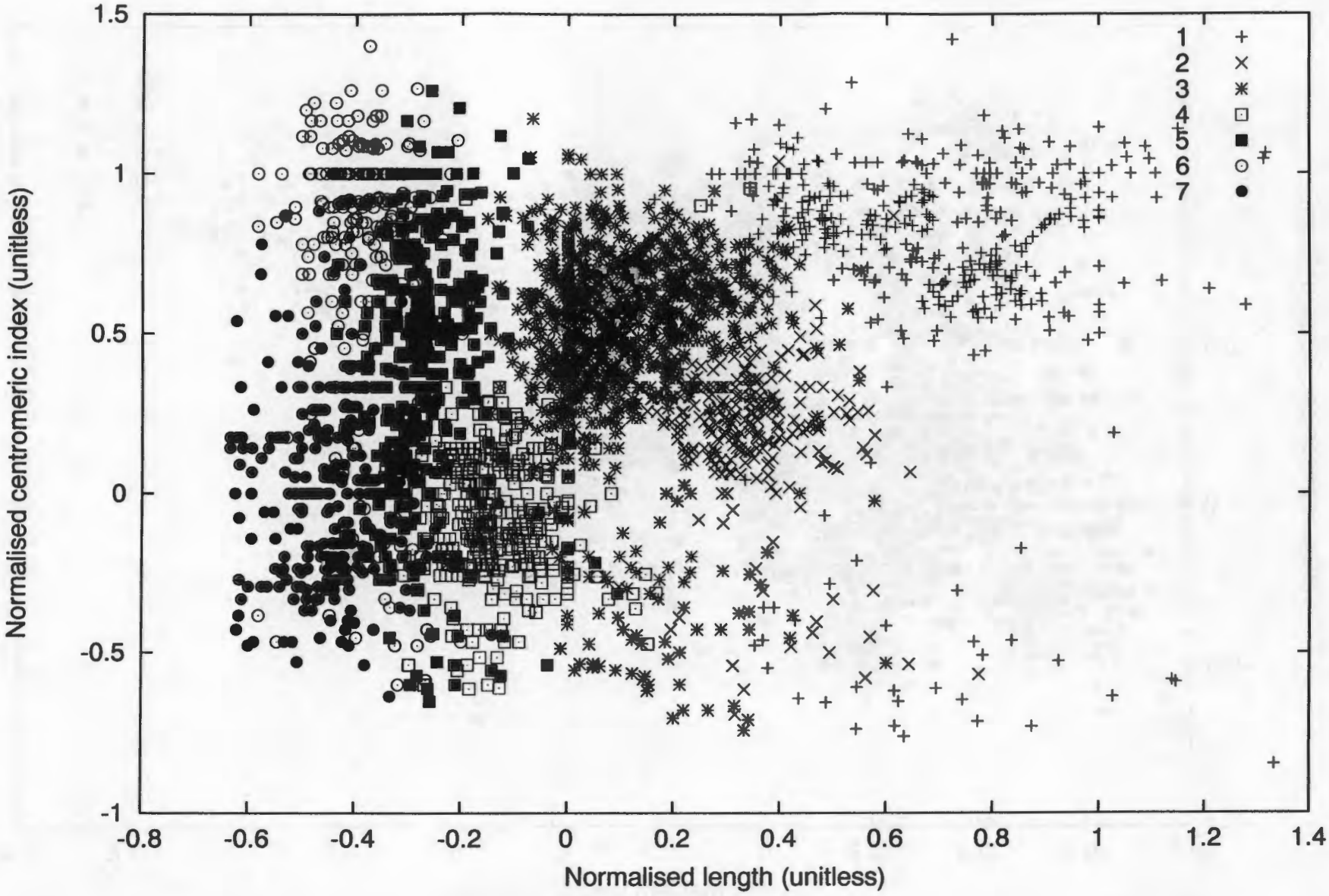


Figure G.3: Plot of normalised centromeric index against normalised length for the first half of the Philadelphia data set.

Bibliography

[1] James A. Anderson and Edward Rosenfeld, editors. *Neurocomputing: Foundations of Research.* MIT Press, Cambridge, Massachusetts, 1988.

[2] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis.* John Wiley, New York, 1958.

[3] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[4] H. D. Block. The perceptron: a model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962. Reprinted in [1].

[5] Ronald N. Bracewell. *The Fourier Transform and Its Applications.* McGraw-Hill, second edition, 1978.

[6] Andrew Carothers and Jim Piper. Computer-aided classification of human chromosomes: a review. *Statistics and Computing*, 4:161–171, 1994.

[7] J.M. Connor and M.A. Ferguson-Smith. *Essential Medical Genetics.* Blackwell Scientific Publications, Oxford, third edition, 1991.

[8] Lawrence Davis, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[9] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, Inc., 1973.

[10] Phil A. Errington and Jim Graham. Application of artifical neural networks to chromosome classification. *Cytometry*, 14:627–639, 1993.

[11] Phil A. Errington and Jim Graham. Classification of chromosomes using a combination of neural networks. In M. Taylor and P. Lisboa, editors, *Techniques and Applications of Neural Networks*, chapter 5, pages 77–92. Ellis Horwood, 1993.

[12] A. Forabosco, P. Battaglia, and R. Bolzani. Density profiles in human chromosome analysis. In C. Lundsteen and J. Piper, editors, *Automation of Cytogenetics*, pages 253–262. Springer Verlag, Heidelberg, 1989.

[13] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, second edition, 1990.

[14] Tommy Gerdes and Claes Lundsteen. Automatic classification of "Copenhagen" chromosomes using weighted density distributions and/or band transition sequences. Private Communication.

[15] J. Gibb. Back propagation family album. Technical Report C/TR96-05, Department of Computing, Macquarie University, August 1996.

[16] David E. Goldberg. *Genetic Algorithms in Search, Optimisation, and Machine Learning.* Addison-Wesley Publishing Company, Inc., 1989.

[17] Ursula Goodenough. *Genetics.* Holt, Rinehart and Winston, Inc., 1974.

[18] Goesta H. Granlund. The use of distribution functions to describe integrated density profiles of human chromosomes. *Journal of Theoretical Biology*, 40:573–589, 1971.

[19] Goesta H. Granlund. Statistical analysis of chromosome characteristics. *Pattern Recognition*, 6:115–126, 1974.

[20] Goesta H. Granlund. Identification of human chromosomes by using integrated density profiles. *IEEE Transaction on Biomedical Engineering*, BME-23(3):182–192, 1976.

[21] Erik Granum. Application of statistical and syntactical methods of analysis and classification to chromosome data. In J. Kittler, K. S. Fu, and L. F. Pau, editors, *Pattern Recognition Theory and Applications*, pages 373–398. D. Reidel, Dordrecht, Netherlands, 1982.

[22] Erik Granum and Michael G. Thomason. Automatically inferred Markov network models for classification of chromosomal band pattern structures. *Cytometry*, 11:26–39, 1990.

[23] John J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-16(1):122–128, 1986.

[24] Frans C.A. Groen, Ton K. ten Kate, Arnold W.M. Smeulders, and Ian T. Young. Human chromosome classification based on local band descriptors. *Pattern Recognition Letters*, 9:211–222, 1989.

[25] J. D. F. Habbema. A discriminant analysis approach to the identification of human chromosomes. *Biometrics*, 32:919–928, 1976.

[26] J. D. F. Habbema. Statistical methods for classification of human chromosomes. *Biometrics*, 35:103–118, 1979.

[27] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation.* Addison–Wesley Publishing Company, 1991.

[28] C. Judith Hilditch and Denis Rutovitz. Normalization of chromosome measurements. *Computers in Biology and Medicine*, 2:167–179, 1972.

[29] A. Hoekstra, M. A. Kraaijveld, D. de Ridder, and W. F. Schmidt. *The Complete SPRLIB & ANNLIB.* Pattern Recognition Group, Delft University of Technology, April 1996.

[30] Dan H. Moore II. Normalisation of chromosome measurements: a new method. *Computers in Biology and Medicine*, 5:21–28, 1975.

[31] Anil K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, Inc., 1989.

[32] Anne M. Jennings and Jim Graham. A neural network approach to automatic chromosome classification. *Physics in Medicine and Biology*, 38:959–970, 1993.

[33] Dennis A. Johnston, K. S. Tang, and Stuart Zimmerman. Band features as classification measures for G-banded chromosome analysis. *Computers in Biology and Medicine*, 23(2):115–129, 1993.

[34] Walter P. Sweeney Jr., Mohamad T. Musavi, and John N. Guidi. Classification of chromosomes using a probabilistic neural network. *Cytometry*, 16:17–24, 1994.

[35] S. P. J. Kirby and C. M. Theobald. Some two-stage procedures for the calculation of discriminant scores in the automated allocation of human chromosomes. *Pattern Recognition Letters*, 14:221–227, 1993.

[36] S. P. J. Kirby, C. M. Theobald, J. Piper, and A. D. Carothers. Some methods of combining class information in multivariate normal discrimination for the classification of human chromosomes. *Statistics in Medicine*, 10:141–149, 1991.

[37] S. Kirkpatrick, C. D. Gelatt, Jr., and M.P. Vecchi. Optimisation by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[38] P. Kleinschmidt, C.W. Lee, and H. Schannath. Transportation problems which can be solved by the use of Hirsch-paths for the dual problem. *Mathematical Programming*, 37:153–168, 1987.

[39] Peter Kleinschmidt, Ilse Mitterreiter, and Jim Piper. Improved chromosome classification using monotonic functions of Mahalanobis distance and the transportation method. *ZOR - Mathematical Methods of Operation Research*, 40:305–323, 1994.

[40] Peter Kleinschmidt, Ilse Mitterreiter, and Christian Rank. A hybrid method for automatic chromosome karyotyping. *Pattern Recognition Letters*, 15:87–96, 1994.

[41] Gert Korthof and Andrew D. Carothers. Tests of performance of four semi-automatic metaphase-finding and karyotyping systems. *Clinical Genetics*, 40:441–451, 1991.

[42] Robert S. Ledley and Frank H. Ruddle. Chromosome analysis by computer. *Scientific American*, 214(4):40–46, April 1966.

[43] B. Lerner, H. Guterman, I. Dinstein, and Y. Romem. Medial axis transform-based features and a neural network for human chromosome classification. *Pattern Recognition*, 28(11):1673–1683, 1995.

[44] David Lloyd, Jim Piper, Denis Rutovitz, and Geoffrey Shippey. Multiprocessing interval processor for automated cytogenetics. *Applied Optics*, 26(16):3356–3366, August 1987.

[45] Claes Lundsteen, Tommy Gerdes, Erik Granum, John Philip, and Kim Philip. Automatic chromosome analysis II. Karyotyping of banded human chromosomes using band transition sequences. *Clinical Genetics*, 19:26–36, 1981.

[46] Claes Lundsteen, Tommy Gerdes, and Jan Maahr. Automatic classification of chromosomes as part of a routine system for clinical analysis. *Cytometry*, 7:1–7, 1986.

[47] Claes Lundsteen, Tommy Gerdes, and Kim Philip. Attributes for pattern recognition selected by stepwise data compression supervised by visual classification. In J. Kittler, K. S. Fu, and L. F. Pau, editors, *Pattern Recognition Theory and Applications*, pages 399–411. D. Reidel, Dordrecht, Netherlands, 1982.

[48] Claes Lundsteen and John Philip. Automated cytogenetic analysis: accomplishments, present status and practical future possibilities. *Clinical Genetics*, 36:386–391, 1989.

[49] Claes Lundsteen, John Philip, and Erik Granum. Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes. *Clinical Genetics*, 18:355–370, 1980.

[50] Timothy Masters. *Practical Neural Network Recipes in C++*. Academic Press, Inc., 1993.

[51] Timothy Masters. *Signal and Image Processing with Neural Networks: A C++ Sourcebook*. John Wiley & Sons, Inc., 1994.

[52] Timothy Masters. *Advanced Algorithms for Neural Networks : A C++ Sourcebook.* John Wiley & Sons, Inc., 1995.

[53] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943. Reprinted in [1].

[54] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker, New York, 1988.

[55] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[56] Marvin Minsky and Seymour Papert. *Perceptrons.* MIT Press, Cambridge, Massachusetts, 1969.

[57] Optimas Corporation, 18911 North Creek Parkway, Suite 101, Bothell, Washington 98011. *Optimas Library User Guide*, first edition, October 1995.

[58] Optimas Corporation, 18911 North Creek Parkway, Suite 101, Bothell, Washington 98011. *OPTIMAS 6 User Guide and Technical Reference*, eighth edition, September 1996.

[59] J. Piper, E. Granum, D. Rutovitz, and H. Ruttledge. Automation of chromosome analysis. *Signal Processing*, 2:203–221, 1980.

[60] Jim Piper. Finding chromosome centromeres using boundary and density information. In J. C. Simon and R. M. Haralick, editors, *Digital Image Processing*, pages 511–518. D. Reidel, Dordrecht, Netherlands, 1981.

[61] Jim Piper. Classification of chromosomes constrained by expected class size. *Pattern Recognition Letters*, 4:391–395, 1986.

[62] Jim Piper. The effect of zero feature correlation assumption on maximum likelihood based classification of chromosomes. *Signal Processing*, 12:49–57, 1987.

[63] Jim Piper. Variability and bias in experimentally measured classifier error rates. *Pattern Recognition Letters*, 13:685–692, 1992.

[64] Jim Piper. Genetic algorithm for applying constraints in chromosome classification. *Pattern Recognition Letters*, 16:857–864, 1995.

[65] Jim Piper and Erik Granum. On fully automatic feature measurement for banded chromosome classification. *Cytometry*, 10:242–255, 1989.

[66] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing.* Cambridge University Press, second edition, 1992.

[67] B. D. Ripley. Flexible non-linear approaches to classification. In V. Cherkassky, J. H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*, pages 105–126. Springer-Verlag, 1994.

[68] B. D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[69] Gunter Ritter, Maria Teresa Gallegos, and Karl Gaggermeier. Automatic context-sensitive karyotyping of human chromosomes based on elliptically symmetric statistical distributions. *Pattern Recognition*, 28(6):823–831, 1995.

[70] F. Rosenblatt. *Principles of Neurodynamics.* Spartan, New York, 1962.

[71] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, volume 1*, pages 318–362. MIT Press, Cambridge, Massachusetts, 1986. Reprinted in [1].

[72] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. Reprinted in [1].

[73] Denis Rutovitz. Chromosome classification and segmentation as exercises in knowing what to expect. In E. W. Elcock and D. Michie, editors, *Machine Intelligence 8*, pages 455–472. Ellis Horwood, 1977.

[74] R. E. Slot. On the profit of taking into account the known number of objects per class in classification methods. *IEEE Transactions on Information Theory*, IT-25(4):484–488, 1979.

[75] Donald Specht. Probabilistic neural networks. *Neural Networks*, 3:109–118, 1990.

[76] A. T. Sumner, H. J. Evans, and R. A. Buckland. New technique for distinguishing between human chromosomes. *Nature New Biology*, 232(27):31–32, 1971.

[77] Charles W. Therrien. *Decision, Estimation and Classification : An Introduction to Pattern Recognition and Related Topics.* John Wiley & Sons, 1989.

[78] Michael G. Thomason and Erik Granum. Dynamic programming inference of Markov networks from finite sets of sample strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4), 1986.

[79] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley, New York, 1985.

[80] J. H. Tjio and A. Levan. The chromosome number of man. *Heriditas*, 42:1–6, 1956.

[81] Michael Tso and Jim Graham. The transportation algorithm as an aid to chromosome classification. *Pattern Recognition Letters*, 1:489–496, 1983.

[82] Michael Tso, Peter Kleinschmidt, Ilse Mitterreiter, and Jim Graham. An efficient transportation algorithm for automatic chromosome karyotyping. *Pattern Recognition Letters*, 12:117–126, 1991.

[83] Lucas J. van Vliet, Ian T. Young, and Brian H. Mayall. The Athena semi-automated karyotyping system. *Cytometry*, 11:51–58, 1990.

[84] Robert L. White. *Basic Quantum Mechanics.* McGraw-Hill Inc., 1966.

[85] Bernard Widrow and Marcian E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, pages 96–104, New York, 1960. IRE. Reprinted in [1].

[86] A.M. Winchester and Thomas R. Mertens. *Human Genetics.* Charles E. Merrill Publishing Company, Columbus, Ohio, fourth edition, 1983.

[87] Stuart O. Zimmerman, Dennis A. Johnston, Frances E. Arrighi, and M. E. Rupp. Automated homologue matching of human G-banded chromosomes. *Computers in Biology and Medicine*, 16:223–233, 1986.

Acknowledgements

I would like to acknowledge the following people who have contributed in various ways to this dissertation and to my enjoyment of life during the preparation of this dissertation.

- My thesis supervisor, Professor Gerald Robertson, for his advice, encouragement, many interesting conversations and provision of up-to-date computer facilities.
- Dr. Jim Piper of Vysis, Inc. for allowing me to use his extensive sets of chromosome data, and Dr. Richard Baldock of the UK MRC Human Genetics Unit for providing me with the data on CD-ROM.
- Dr. Claes Lundsteen and Dr. Tommy Gerdes of the Rigshospitalet in Copenhagen for providing me with the original versions of the Copenhagen data sets.
- The Natal Institute of Immunology and specifically Dr. Jan Conradie for giving me the opportunity to implement a karyotyping system.
- The UCT Physics department for giving me the freedom to completely redesign the PHY104W laboratory course and teach it in my own style, and to Dr. Saalih Allie and Dr. Loveness Kaunda for much advice and help in the redesign of this course.
- Dr. Andrew Stoddart and the University of Surrey Electrical and Electronic Engineering Department for hosting me on my two month visit.
- Optimas Corporation in Seattle for hosting me for a two week visit.
- Ira Haron (originally at Optimas Corporation) for many interesting discussions on image analysis software, for showing such enthusiasm for and interest in my karyotyping software, and for providing a large number of suggestions for improving the software.
- The Foundation for Research Development (FRD) and the University of Cape Town for financial support.
- My parents for footing the bill for my first four years of university, and for their continuous support.

- My flat-mates during the time it took me to prepare this thesis: Kerry-Anne Hare, Jarrod Hart, Tim Hawkins, Anthony Knobel, Antoine de Rodellec and Greg Torr.
- My friends David Brookes, Bronwyn Butcher, Justin Templemore-Finlayson, Narendra Viranna and Mark Marais.