Journal of Biomedical Informatics 42 (2009) 22–31

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Automated classification of metaphase chromosomes: Optimization of an adaptive computerized scheme

Xingwei Wang [a], Bin Zheng [b], Shibo Li [c], John J. Mulvihill [c], Marc C. Wood [a], Hong Liu [a,*]

[a] *Center for Bioengineering and School of Electrical and Computer Engineering, University of Oklahoma, 202 West Boyd Street, Room 219, Norman, OK 73019, USA*
[b] *Department of Radiology, University of Pittsburgh Medical Center, Pittsburgh, PA 15213, USA*
[c] *Department of Pediatrics, University of Oklahoma Health Science Center, Oklahoma City, OK 73104, USA*

ARTICLE INFO

Article history:
Received 16 October 2007
Available online 21 May 2008

Keywords:
Artificial neural network
Genetic algorithm
Karyotype
Metaphase chromosome
Training–testing–validation

ABSTRACT

We developed and tested a new automated chromosome karyotyping scheme using a two-layer classification platform. Our hypothesis is that by selecting most effective feature sets and adaptively optimizing classifiers for the different groups of chromosomes with similar image characteristics, we can reduce the complexity of automated karyotyping scheme and improve its performance and robustness. For this purpose, we assembled an image database involving 6900 chromosomes and implemented a genetic algorithm to optimize the topology of multi-feature based artificial neural networks (ANN). In the first layer of the scheme, a single ANN was employed to classify 24 chromosomes into seven classes. In the second layer, seven ANNs were adaptively optimized for seven classes to identify individual chromosomes. The scheme was optimized and evaluated using a "training–testing–validation" method. In the first layer, the classification accuracy for the validation dataset was 92.9%. In the second layer, classification accuracy of seven ANNs ranged from 67.5% to 97.5%, in which six ANNs achieved accuracy above 93.7% and only one had lessened performance. The maximum difference of classification accuracy between the testing and validation datasets is <1.7%. The study demonstrates that this new scheme achieves higher and robust performance in classifying chromosomes.

1. Introduction

Since Tjio and Levan discovered that the number of human chromosomes was 46 in 1956 [1] and the Denver group classification standard was established in 1960 [2], karyotyping of human chromosomes has became an important clinical procedure for screening and diagnosing genetic disorders and cancers [3]. Karyotyping is a standard technique utilized to classify metaphase chromosomes into 24 types. Fig. 1 demonstrates a male normal metaphase spread and the corresponding karyotype of chromosomes. Because manual karyotyping is a labor-intensive and time-consuming task, developing automatic computer-assisted karyotyping systems has attracted significant research interest for the last 30 years [4].

In the development of automated karyotyping schemes, the extraction and computation of chromosome image features as well as the selection and optimization of feature classifiers are two most important challenges. Due to banding patterns of the metaphase chromosomes, many features related to global and local banding characteristics, chromosome length, and centromere index (CI) have been extracted and computed in the previous studies [5–8]. While the banding features were computed from the chromosome density profiles in most of the studies [9], the wavelet-based banding features were also tested in other study [10]. Since there are no established standards (or commonly accepted rules) to compute and select image features, many of initially computed features can be redundant. Thus, feature selection is a vital process for identifying chromosomes and a small set of features can significantly affect the accuracy and efficiency of the chromosome classification [11,12]. Researchers have tried and tested different methods to select optimal feature sets to represent chromosomes. For examples, one study implemented the "knocking-out" algorithm to select features from density profiles, CI and chromosome lengths [13], and another study applied principle component analysis (PCA) and discrete cosine transform (DCT) functions to define and identify features that have higher discrimination power to classify chromosomes [14].

In order to automatically classify metaphase chromosomes, different classifiers have also been investigated and reported in previous studies, which include statistical models [3,5,6,8,10,14,15], artificial neural networks (ANN) [9,13,16–21], knowledge-based expert schemes [22–24], transportation algorithm [12], homologue matching algorithm [25], the fuzzy-logic based classifier [26], and other methods [27–29]. Among them, statistical algorithms and ANN classifiers are the most popular methods and studies have

[*] Corresponding author. Fax: +1 405 325 7066.
E-mail address: liu@ou.edu (H. Liu).

1532-0464/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2008.05.004

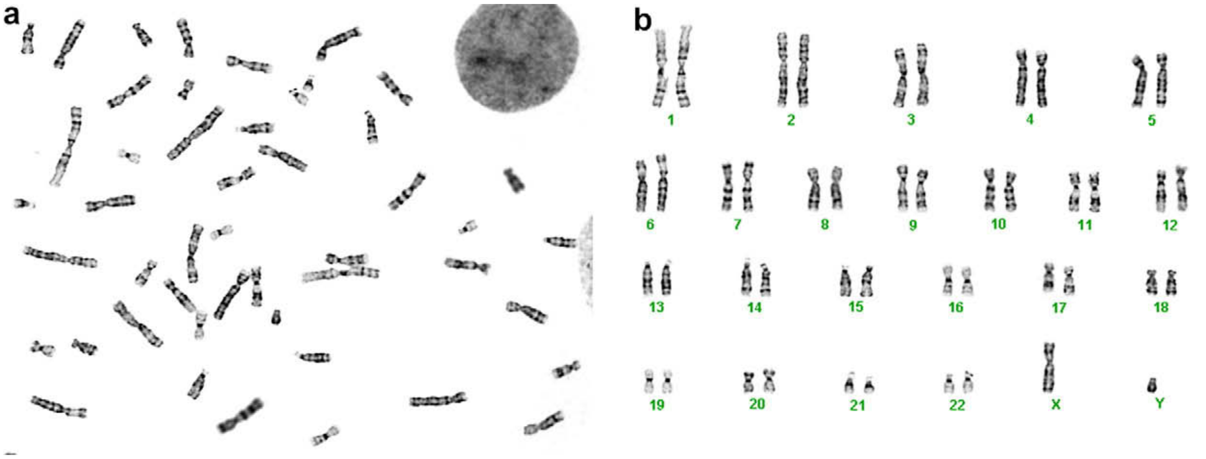

Fig. 1. (a) A metaphase spread image and (b) the corresponding karyotype image.

showed that both types of classifiers yield comparable results when classifying human chromosomes [30]. One study showed that an ANN and a maximum likelihood (ML) based classifier achieved accuracy rates of 82.8% and 81.7%, respectively, when applying to the same database [18]. The main advantages of ANN include that (1) it is capable of modeling the human brain ability to recognize objects based on incomplete or partial information and (2) it is relatively easy to be trained because of its simple topographic structure [31]. As a result, several research groups have developed and tested different ANNs for the classification of metaphase chromosomes. In most of the these studies, a single large size ANN was developed to classify all of 24 types of chromosomes, while publicly available databases (i.e., Copenhagen, Edinburgh, and Philadelphia) and a jackknifing (leave-one-out) method were used to train and test the ANN [9,13,16,17,19]. For examples, the first group trained and tested three ANNs with 15 input neurons, three different hidden neurons (10, 15, and 20), and 23 output neurons to classify 23 types of chromosomes (omitting chromosome Y). The study reported the average classification error rate as 10.3% on the Copenhagen dataset [9]. The second group developed and tested an ANN with 15 input neurons, 100 hidden neurons, and 24 output neurons and it reported that classification error rates were 6.2%, 17.8%, and 22.7% for the Copenhagen, Edinburgh, and Philadelphia databases, respectively [17]. The third group trained an ANN with 27 input neurons and reported the classification error rate of 6.52% on the Copenhagen Dataset [19].

Despite of the research efforts and progress made in the previous studies, these ANN-based classifiers have a number of limitations. First, developing a single ANN to simultaneously classify 24 types of chromosomes makes the classifier complicated and difficult to train [24]. It also tends to generate unstable results. A previous study showed that by reducing the size of a single ANN, the testing accuracy on chromosome classification increased from 75.8% to 88.3% [20]. Second, the number of input neurons and hidden neurons was all empirically selected resulting in large variations among different ANNs when applied to the same public databases. Third, a large and complex ANN needs to be trained using a large size dataset in order to achieve robust results. Although a leave-one-out method takes full advantage of the database by using the maximum number of training data, it has two disadvantages including that (1) it requires the high computational cost in ANN training since it needs to train ANN *N* times instead of training once for a database containing *N* chromosomes and (2) it cannot generate a single optimal and workable ANN for future testing [32]. Finally, the robustness of these ANNs has not been evaluated using independent validation dataset.

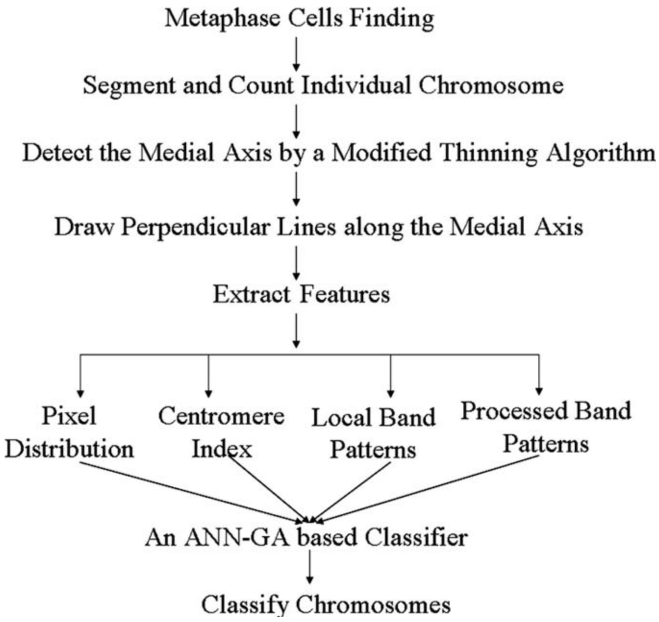
Metaphase Cells Finding
Segment and Count Individual Chromosome
Detect the Medial Axis by a Modified Thinning Algorithm
Draw Perpendicular Lines along the Medial Axis
Extract Features
Pixel Distribution
Centromere Index
Local Band Patterns
Processed Band Patterns
An ANN-GA based Classifier
Classify Chromosomes

Fig. 2. A flow diagram of automated classification of chromosomes.

The motivation of this study is to investigate a new approach to overcome the limitations of previous approaches to optimize ANNs for classification of chromosomes. Our hypothesis is that by selecting most effective or optimal feature sets and adaptively optimizing a sets of small size ANN classifiers, we can reduce the complexity of automated karyotyping scheme and improve its performance and robustness. To test this hypothesis, we proposed to develop and test a new computerized scheme as shown in Fig. 2. In this study, we focused our research effort on identifying effective image features, adaptively optimizing ANN classifiers for different groups of chromosomes, and testing scheme performance and robustness. The detailed description of the scheme development and the experimental results is presented in the following sections.

2. Materials and methods

2.1. An experimental database

In this study, we selected 150 various metaphase chromosome cells, which were originally obtained from peripheral blood and amniotic fluid samples of patients, who underwent diagnosis at the genetic laboratory of the University of Oklahoma Health Science Center (OUHSC). All metaphase cells were stained with Giemsa dye mixture as the staining agent, and the band levels of these chromosomes were determined to be 400. Fig. 3(a) shows chromosome #1 with 400 bands. The digital images of the metaphase chromosomes were captured using a digital camera installed on the Nikon LABOPHOT-2 optical microscope, which is equipped with an oil immersion based objective for magnification of 100× and having a numerical aperture (NA) of 1.45. The pixel size is 0.2 μm × 0.2 μm on the sample slides. A computer scheme was applied to detect and identify analyzable metaphase chromosome cells depicted on acquired digital images. It first uses a median filter to reduce the image noise. After applying an adjustable threshold to segment initially suspicious chromosomes, the scheme uses a component labeling algorithm and a raster scanning method to label and group the segmented regions and delete the isolated small areas. The scheme then computes a set of features from the segmented regions and applies a decision-tree based classifier to identify analyzable metaphase cells [33]. In this study, the computerized identified analyzable metaphase cells were visually examined and confined by an experienced cytogeneticist. The selected 150 metaphase cells were then randomly divided into three independent training, testing and validation datasets, each of which includes 50 metaphase cells and 2300 individual chromosomes. Specifically, each dataset includes 100 chromosomes for each of 22 types (from chromosome #1 to #22). In addition, the training dataset includes 62 X and 38 Y chromosomes, while testing and validation datasets include 64 X and 36 Y, and 63 X and 37 Y chromosomes, respectively.

2.2. Feature computation

To extract and compute chromosome image features, we first applied a modified thinning algorithm to detect the medial axis for the chromosome [34]. During this detection process, a conventional thinning algorithm is first applied to detect the initial medial axis, in which some pixels near both ends of a chromosome are missing and some redundant pixels are generated around the middle section of the chromosome. Second, an interpolation algorithm is followed to connect every selected fifth pixel and generate a new smoothed medial axis that can delete the redundant pixels. To retrieve the missing pixels near both end of the axis, the algorithm searches for the tip pixels based on the extension (interpolation) of previous slopes of the ending pixels of the medial axis. The revised medial axis is then connected based on the smoothed slopes of every pair of the selected fifth pixels. Finally, the algorithm checks whether the ending pixels reach the exterior contour of the chromosome; if they do, the procedure is completed and an “optimal” medial axis is detected. Otherwise, the algorithm iteratively retraces two ending pixels at the medial axis until they reach the exterior contour of the chromosome. Fig. 3(b) displays a number of detected medial axes of the chromosome #1 with different morphologies. The more detailed description of this detection algorithm and experimental results has been previously reported [34].

A computer scheme was applied to compute chromosome features. For each chromosome, 31 features are computed to form an initial feature pool, which is listed and categorized in Table 1. As shown in Table 1, four categories of features are extracted and computed for each chromosome. To extract these features, three image profiles including density, shape, and banding profile are calculated [34]. Each profile defines a one-dimensional graph of a chromosome property computed at a sequence of points along the identified medial axis of a chromosome.

(1) A density profile determines the average grey scale value of every perpendicular line across the medial axis of a chromosome (x). It is computed as: $D(x) = [\sum_{i=1}^{n} g_i(x)]/n$, where $g_i(x)$ *is the gray value of each pixel in a perpendicular line, and n* is the number of all pixels in each perpendicular line. The computer scheme applies a median filter to reduce possible impulses and noise in the density profile. Fig. 4(a–c) are the corresponding density profiles for chromosome #22, #10, and #1, respectively.

(2) A shape profile records the weighted width of every perpendicular line across the medial axis of a chromosome (x). It is defined as: $S(x) = \sum_{i=1}^{n}[g_i(x) \times d_i(x)^2]/\sum_{i=1}^{n} d_i(x)^2$, which corresponds to the sum of the product of the grey scale value $g_i(x)$ and its corresponding Euclidean distance $d_i(x)$ away from the medial axis of the perpendicular line, divided by the sum of the distance [6]. Fig. 4(d–f) describes the shape profile of chromosome #22, #10, and #1, respectively.

(3) An idealized banding profile is computed by processing a density profile $D(x)$ with a non-linear transform filter defined by Kramer and Bruckner method [35]. It is a profile in which each band is characterized by a uniform density and the transitions between neighboring bands are step functions [36]. By assuming that x is the index number of a profile, $B(x)$ is an original banding profile obtained by a

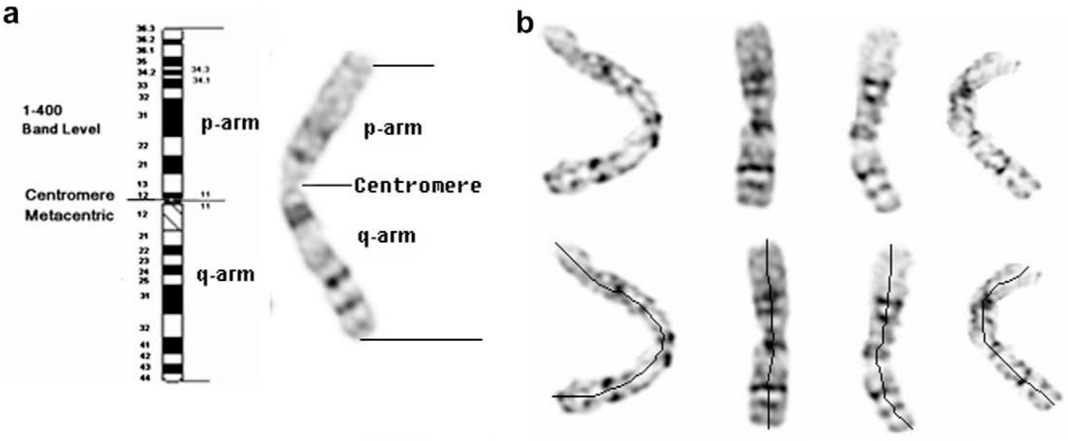

Fig. 3. (a) Illustration of an ideogram of chromosome #1 and a real chromosome #1, (b) several medial axis detection results of chromosome #1 with different morphologies obtained by a modified thinning algorithm.

Table 1
Distribution of 31 computed chromosome features in the initial feature pool

Feature category	Number of features	Brief feature description
1. Pixel distribution	3	Chromosome size, length, and average density.
2. Centromere index (CI)	2	Area and length of CI.
3. Local band patterns	12	Band distributions computed from the original chromosome image.
4. Processed band patterns	14	Band patterns computed from 8 WDD functions and 6 differences (the first order derivatives) of WDD functions.

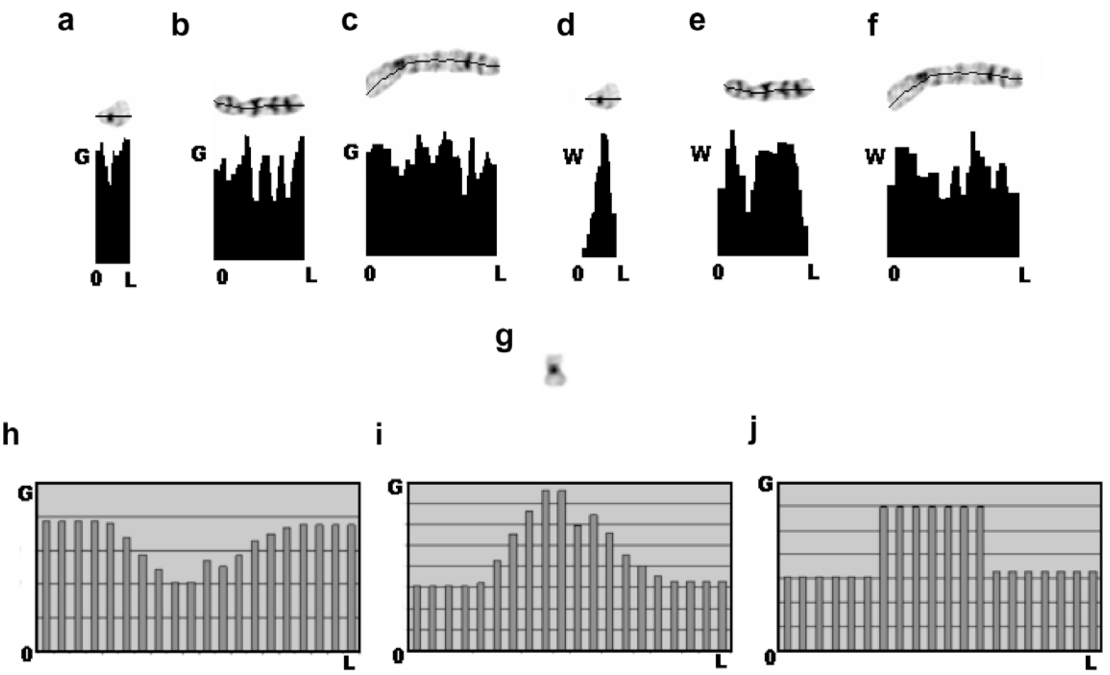

Fig. 4. (a) A density profile of chromosome #22, (b) a density profile of chromosome #10, (c) a density profile of chromosome #1, (d) a shape profile of chromosome #22, (e) a shape profile of chromosome #10, (f) a shape profile of chromosome #1, (g) an example of chromosome #19, (h) an original banding profile, (i) a reversed banding profile, (j) an idealized density profile gained by a non-linear file. *Note:* G is the gray value of all pixels in each perpendicular line across the medial axis of a chromosome; *W* is the width of a chromosome; *L* is the length of a chromosome.

median filtered density profile, IB(x), is an idealized banding profile, NF[$B(x)$] is a non-linear filter for $B(x)$, and $N(x)$ is neighborhood of $B(x)$ the scheme computes:

$$N(x) = [B(x-1), B(x), B(x+1)] \tag{1}$$

$$\mathrm{DIFMAX} = \mathrm{MAX}[N(x)] - B[x] \tag{2}$$

$$\mathrm{DIFMIN} = B[x] - \mathrm{MIN}[N(x)] \tag{3}$$

$$\mathrm{IB}(x) = \mathrm{NF}[B(x)] = \begin{cases} B(x) + \mathrm{DIFMAX}/R & \text{If} \quad \mathrm{DIFMAX} \leqslant \mathrm{DIFMIN} \\ B(x) - \mathrm{DIFMIN}/R & \text{If} \quad \mathrm{DIFMIN} < \mathrm{DIFMAX} \end{cases} \tag{4}$$

An iterative computing method is applied to identify the optimal IB(x). In the first iteration, $R = 2$, while in the following iteration steps, $R = 1$. The iterations are configured to continue until the result of the previous step is the same as the current iteration. The idealized banding profile can avoid the transitions between black and white bands and reduce errors of analyzing band features. For example, Fig. 4(g) is an example of chromosome #19 and Fig. 4(h–j) show an original banding profile, a reversed banding profile, and an idealized banding profile gained by a non-linear filter, respectively.

As shown in Table 1, 31 features can be divided into four categories. The first category includes three features representing pixel distribution of each chromosome. These are (1) size that is determined by counting the number of chromosome pixels, (2) length that is defined by counting the number of pixels in the full skeleton of a chromosome, and (3) density that is computed as average gray value of all chromosome pixels.

The second category includes the centromere index related features. A centromere is a unique region in the chromosome where the chromatids are joined and by which the chromosome is attached to the spindle during cell division [37]. A centromere separates a chromosome into two arms a shorter arm (p-arm) and a longer arm (q-arm) as shown in Fig. 4(a). Polarity assignment determines the orientation of a chromosome through the identification of a p-arm and a q-arm. In our previous study, we applied a computing algorithm to identify centromeres and polarities [34]. A rule-based classification approach is applied to search for the global minimum in different ranges of shape profiles as well as detect centromere and assign polarities of chromosomes with various sizes. The centromere index (CI) is also computed as the ratio of the length of a shorter arm to the total length of the chromosome. Thus, two features in this category are computed as (1) area of CI, denoted by $\mathrm{CI}(A)$=$A_\mathrm{p}/(A_\mathrm{p}$+$A_\mathrm{q})$ where A_p is the area of a p-arm and A_q is the area of a q-arm, and (2) length of CI, denoted by $\mathrm{CI}(L) = L_\mathrm{p}/(L_\mathrm{p} + L_\mathrm{q})$ where L_p is the length of a p-arm and L_q is the length of a q-arm.

The third feature category contains 12 local band related features. The scheme segments the band pattern and identifies corresponding banding features by computing the first and second derivatives of an idealized banding profile. The scheme scans the first and second derivative profiles. During the process, the scheme searches for points (pixels) in which the first derivative is zero indicating a transition point of the band. The scheme then checks the second derivative values of these points. If the value of the second derivative is negative, this is the local maximum point representing a peak of the dark band. Otherwise, this point is the local minimum representing a valley of the white band. We identify black bands as the areas between two peaks and white bands as the areas between two valleys. Based on the scanning results, the scheme examines four banding characteristics involving (a) the band mass that is the number of pixels in the band, (b) the band position that is the location of the peak of a black band or the valley of a white band, (c) the band width that is the length between pixels of two peaks or valleys, and (d) the band height that is the gray value of the peak (or valley) in the band. Based on these characteristics, the scheme computes 12 features, which includes (1) the average pixel value of the darkest band in a chromosome, (2) the location of the darkest band, (3) the average pixel value of the centromere line, (4) the location of the first black band, (5) the ratio of the largest white area to the total chromosome area, (6) the total number of detected bands in a chromosome, (7) the number of bands on a p-arm, (8) the number of bands on a q-arm, (9) the number of black bands on a p-arm, (10) the number of black bands on a q-arm, (11) the total number of black bands in the chromosome, and (12) the total number of white bands in a chromosome.

The fourth feature category includes 14 image features, which are computed from global band pattern and based on the weighted density distribution (WDD) functions [36]. WDD is defined as: $\mathrm{WDD}_j(x) = \frac{\sum_{i=1}^{n} W_{ji}(x) \times D_i(x)}{\sum_{i=1}^{n} D_i(x)}$, where $j = 1,2,....,8$, $D_i(x)$ is the density value of a density profile $D(x)$ of a chromosome (x) with length of n points (pixels) and $W_{ji}(x)$ is the *jth weighted* function value at the sample point (i) along the medial axis of each chromosome. Eight different weight functions are demonstrated in Fig. 5. WDD analyzes how the density is distributed in a whole chromosome and describes the global band patterns. Six of eight WDD functions (WDD_1 to WDD_6) have been tested in a previous study [6]. In this study, we designed two new functions WDD_7 and WDD_8. Among these WDD functions, WDD_2, WDD_4, and WDD_6 do not depend on the prior knowledge of centromeres and polarities, while WDD_1, WDD_3, and WDD_5 are polarity-dependent functions, WDD_7 and WDD_8 depend on both centromere measurement and polarity assignment. After computation, WDD_1 shows whether the density is mainly distributed in the q-arm of a chromosome, WDD_2 determines whether the density is mainly distributed in the middle of the profile, WDD_3 to WDD_6 mainly calculate the overall density distribution of a chromosome, WDD_7 searches for a dark band in the p-arm of a chromosome, and WDD_8 detects if there are three equally spaced dark bands in the q-arm of a chromosome. Applying the first six weighted functions (seen in Fig. 5) to the $\mathrm{DD}(x)$ which is used to compute the absolute differences of density profile $D(x)$ of a chromosome (x): $\mathrm{DD}_i(x)=|D_i(x)\text{-}D_{i\text{-}1}(x)|$ [6]. The scheme computes six new features:

$$\mathrm{DWDD}_j(x) = \frac{\sum_{i=1}^{n-1} W_{j(i+1)}(x) \times \mathrm{DD}_i(x)}{\sum_{i=1}^{n-1} \mathrm{DD}_i(x)},$$

where $j = 1,2,\ldots,6$, $\mathrm{DD}_i(x)$ is the different density value of $\mathrm{DD}(x)$ at the sample point *(i)* of a chromosome (x), $W_{j(i+1)}(x)$ is the *jth weighted* function value at the sample point $(i + 1)$ along the medial axis of each chromosome. In summary, $WDD_j(x)$ is used to test the density distribution of a chromosome, while $DWDD_j(x)$ is applied to test the absolute difference (similar to the first derivative) of density functions of the chromosomes. These two sets of functions generate total 14 features in this category to describe global band patterns of a chromosome.

Since different metaphase stages can influence the size and band patterns of chromosomes in different cells [36] and the previous study proved that the normalization of size features within each metaphase chromosome cell substantially improved the classification accuracy for karyotyping chromosomes [6], we adaptively normalized each of 31 features within the individual metaphase cell. We assume that a metaphase cell contains N chromosomes (i.e., $N = 46$) and each chromosome includes 31 features. We also define each feature as $F_{ij}, i = 1,...,N$; $j = 1,\ldots,31$ and a feature vector of each chromosome as F_i.For a specific jth feature, the scheme sorts the feature values of N chromosomes in a metaphase

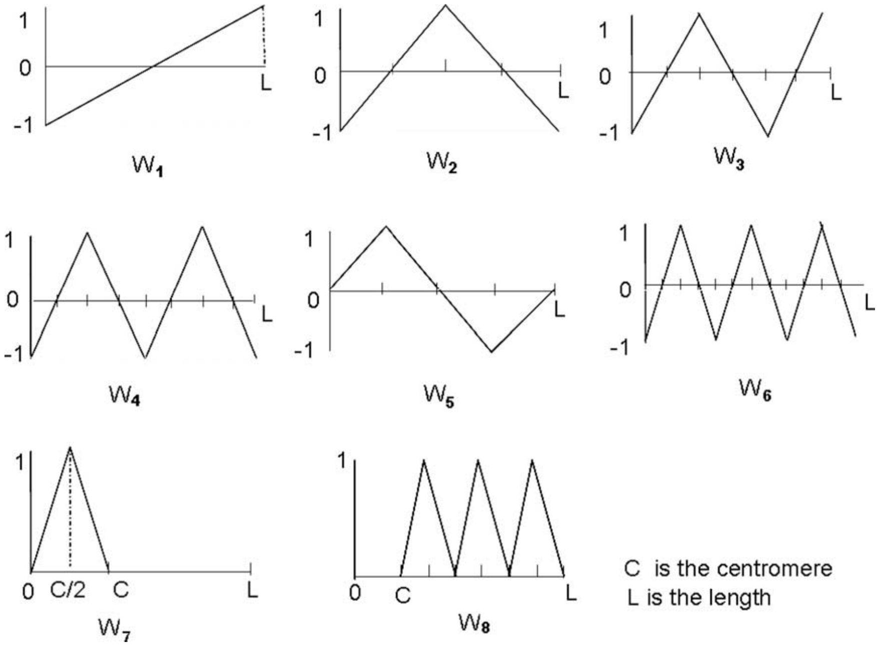

Fig. 5. Display of eight weighted functions.

cell to determine the maximum feature value ($F_j^{\max}$) and the minimum value $F_j^{\min}$. The normalized feature is then computed as:

$$F_{ij}^{\text{norm}} = \frac{F_{ij} - F_j^{\min}}{F_j^{\max} - F_j^{\min}}, \quad i = 1, \ldots, N; j = 1, \ldots, 31.$$

As a result, all feature values representing chromosomes at different metaphase cells are normalized and distributed in the range from 0 to 1.

3. Optimization of a two-layer chromosome classifier

In this study, we built a multi-feature ANN based metaphase chromosome classifier. Since using a single large size ANN to simultaneously classify 24 different types of chromosomes (1–22, X and Y) is a difficult task, which makes the ANN very complex and requires a large number of training samples to achieve a high level of performance, we used a different approach to overcome this difficulty. Our classifier includes a two-layer structure (similar to a decision-tree) to separate a large size ANN into eight small size ANNs (as shown in Fig. 6). Each ANN uses a feed-forward structure with three layers, which includes the input, hidden, and output neurons. Based on Denver group classification standard, a single ANN is implemented to classify chromosomes into seven sub-groups in the first layer of the classifier. These seven sub-groups are named as groups A to G and their common characteristics are described in Table 2. The ANN implemented in the first layer (ANN-1 as shown in Fig. 7(a)) has three output neurons and uses the binary coding method to generate seven sub-groups (as shown in Fig. 7(b)). In the second layer of the classifier, seven ANNs (ANN-2-1 to ANN-2-7) were adaptively optimized to classify the chromosomes into 24 types based on the corresponding subset of training data. After the ANN in the first layer classifies a testing chromosome into one of the seven sub-groups, the second ANN optimized for each sub-group is automatically applied to further classify the chromosome into one type of chromosomes included in this sub-group. For example, if a testing chromosome is assigned to sub-group 1 by ANN-1, ANN-2-1 (as shown in Fig. 6) is then applied. ANN-2-1 has two output neurons. Depending on the output value of ANN-2-1, this testing chromosome is classified as one of the three possible chromosomes covered by this sub-group. Specifically, the ANN output of “00” “01”, and “11” represents chromosome 1, 2, and 3, respectively.

Table 2
The classification of chromosomes based on Denver group classification

Chromosome Class	Size	Relative Position of Centromere[a]
Group A (1–3)	Large	Metacentric
Group B (4–5)	Large	Submetacentric
Group C (6–12,X)	Medium	Submetacentric
Group D (13–15)	Medium	Acrocentric
Group E (16–18)	Relatively short	Submetacentric
Group F (19–20)	Short	Metacentric or Submetacentric
Group G (21–22,Y)	Short	Acrocentric

[a] *Note:* Metacentric-the centromere locates in the middle section of a chromosome, submetacentric-the centromere lies between the middle and the end of a chromosome, and acrocentric-the centromere is close to the end of a chromosome.

We used genetic algorithm (GA) to optimize each ANN by selecting an optimal set of features from our initial pool of 31 features and determine the appropriate number of hidden neurons. GA was first pioneered by John Holland based on the principle of natural selection and population genetics more than 30 years ago [38] and it has been widely studied and implemented in many fields including optimization of the ANN applied in computer-aided detection and diagnosis schemes for medical images [39]. In this study, the binary coding method is applied to create a *chromosome* string used in GA. Note that the *chromosome* used in GA is totally different from the metaphase chromosomes to be classified in this study. To avoid confusion, the GA *chromosomes* are all presented in *Italic format*. Each of the initially extracted 31 features corresponds to a gene of a GA *chromosome*. To determine the number of hidden neurons in the second layer of the ANN, we added four genes in the GA *chromosome*. Thus, the GA *chromosome* string has a fixed length of 35 genes. The first 31 genes represent computed features, in which “1” indicates that the feature represented by this gene is selected or activated, and “0” indicates that the corresponding feature is discarded or inactivated. The last 4 genes of the GA *chromosome* indicate the number of hidden neurons, which is based on the conversion between decimal and binary number

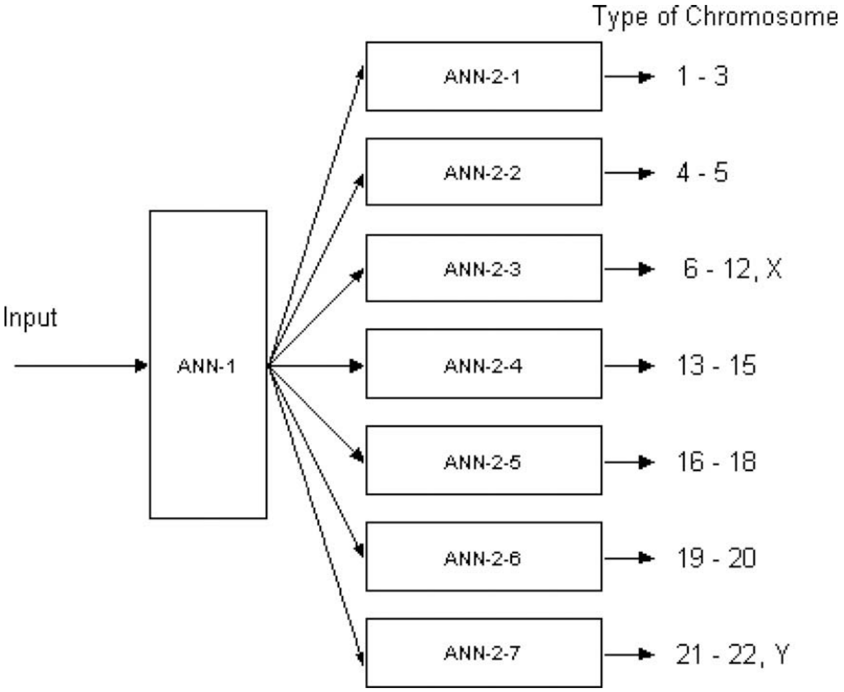

Fig. 6. Illustration of an ANN-based decision tree classifier used to classify chromosomes.

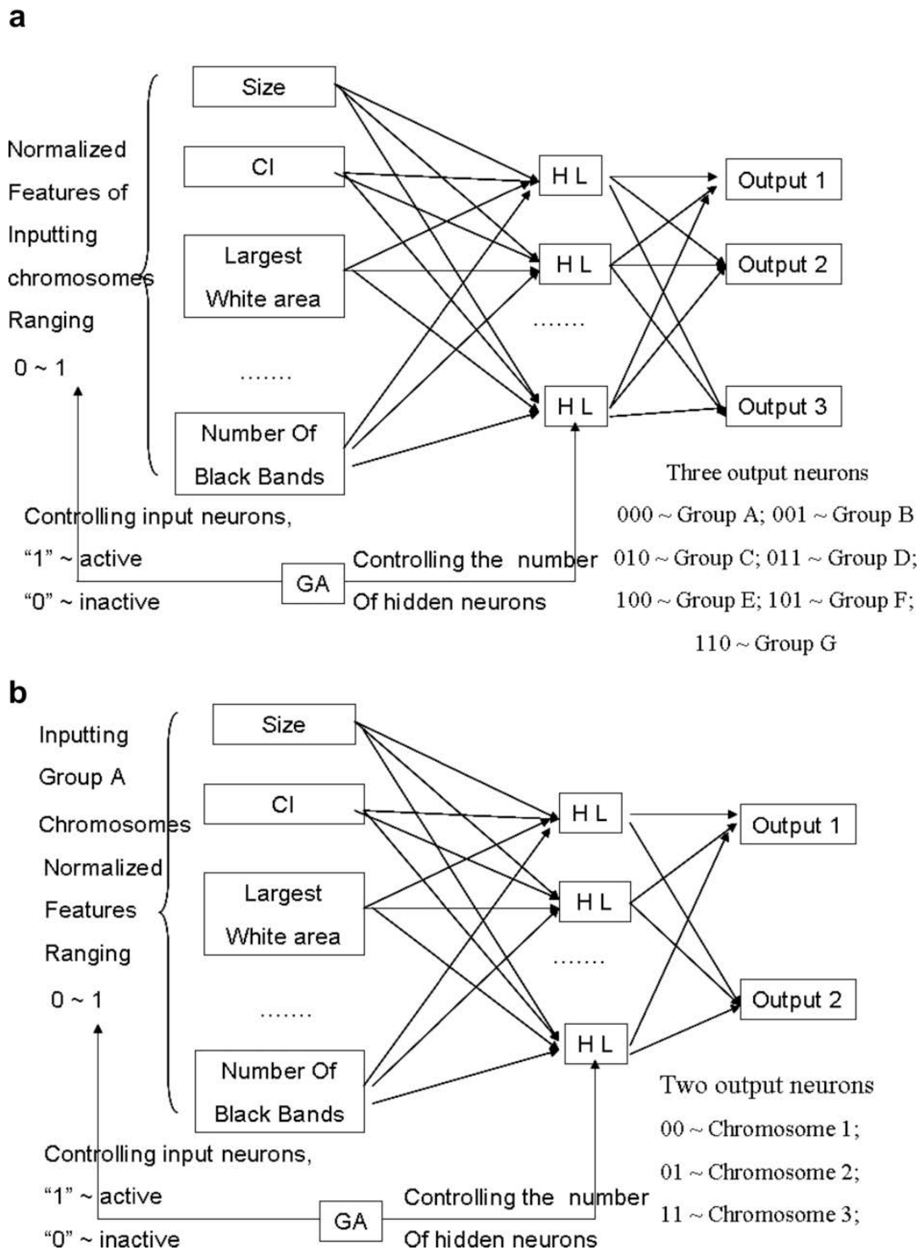
a
Normalized Features of Inputting chromosomes Ranging 0 ~ 1
Size
CI
Largest White area
....
Number Of Black Bands
H L
H L
....
H L
Output 1
Output 2
Output 3
Three output neurons
000 ~ Group A; 001 ~ Group B
010 ~ Group C; 011 ~ Group D;
100 ~ Group E; 101 ~ Group F;
110 ~ Group G
Controlling input neurons,
"1" ~ active
"0" ~ inactive
GA
Controlling the number Of hidden neurons
b
Inputting Group A Chromosomes Normalized Features Ranging 0 ~ 1
Size
CI
Largest White area
....
Number Of Black Bands
H L
H L
....
H L
Output 1
Output 2
Two output neurons
00 ~ Chromosome 1;
01 ~ Chromosome 2;
11 ~ Chromosome 3;
Controlling input neurons,
"1" ~ active
"0" ~ inactive
GA
Controlling the number Of hidden neurons

Fig. 7. (a) Illustration of an ANN optimized by GA in the first layer, (b) illustration of an ANN optimized by GA in the second layer.

systems. For example, "0101" denotes that the ANN has 5 hidden neurons, and "1000" represents the 8 hidden neurons.

A publicly available GA software known as Genesis developed by John J. Grefenstette [40] was selected for this study. To apply the GA software in our optimization application, we designed and coded a new GA evaluation function. After GA generates a testing *chromosome* string that determines the number of ANN input and hidden neurons. The evaluation function calls the pre-developed ANN training and testing programs. Each ANN is trained using the back propagation (BP) method, which operates by learning the weights for a multi-layer network with a fixed set of units and interconnections. Each ANN utilized gradient descent to minimize the squared error between the network output values and the target output values for those outputs [31]. After training and testing datasets are separately used to optimize ANN weights and test ANN performance. The program computes a mean square error (MSE) between ANN generated testing scores and the pre-recorded truth (reference) for all testing samples. The computed MSE is used as a fitness criterion to assess ANN classification performance and it returns to the GA main program. The GA *chromosomes* strings with a smaller MSE have higher probability of being selected to produce new GA *chromosomes* in the next generation using the method of crossover and mutation. GA searches for the best genes to form a new generation of GA *chromosomes*. GA optimization is terminated when the output converges to the smallest MSE value or reaches a predetermined number of generations (i.e., 100). The "best" GA *chromosome* is then used to determine the topology of the ANN applied in our chromosome classifier, which includes the selected features used as the input neurons and the number of hidden neurons.

To generate the highly performed and robust ANNs using GA optimization process, several measures were taken to set up the training or optimization parameters used in ANN training and GA optimization. Since the steepness of the activation function, the learning rate, the momentum term, the learning iterations, and the upper bound of training error can all influence ANN performance and robustness, we implemented two primary procedures to minimize the over-fitting and improve the robustness of the ANN performance. First, we limited the number of training iterations to 1000 and used a large ratio between the momentum (0.8) and learning rate (0.01). Second, for each GA *chromosome*, one ANN is trained with the training dataset and the performance of this ANN is evaluated using an independent testing dataset. Adding this testing method to GA optimization avoids the probability of GA converging to the over-fitted ANNs by always selecting the best trained GA *chromosomes* to generate new GA *chromosomes* in the next generation of optimization. The similar training protocol has been extensively tested in our previous study [39].

Before running GA program, we also need to select a number of initial parameters that include the initialization, evaluation, recombination, crossover, and mutation values of the *chromosomes*. Based on the recommendation of GA software developer [40] and our previous experience [39], we set the initial population size of GA *chromosomes* as 100. We specified the crossover rate, mutation rate and generation gaps to be 0.6, 0.001, and 1.0, respectively. A crossover rate of 0.6 denotes that 60% of genes between two adjacent *chromosomes* are exchanged to produce two new *chromosomes* in the new generation. The selected mutation rate indicates that every gene in the population has 0.1% chance to mutate. Finally, the entire population of *chromosomes* is replaced in each generation, which is represented by a value of generation gap 1.0.

Although GA optimization involves the use of two independent datasets (“training” and “testing”) to minimize the risk of ANN over-fitting, the “testing” dataset is also involved in the optimization cycle. To eliminate the assessment bias, we use another independent dataset (validation dataset) that has not been involved in GA optimization, in an effort to objectively evaluate the performance and robustness of this chromosome classification scheme. The validation performance is tabulated and analyzed.

3. Results

Table 3 summarizes the classification results which include the best GA *chromosome* string for each of eight ANNs and the classification accuracy for both the testing and validation datasets. In the first layer of the classifier, the best GA *chromosome* that represents the ANN-1 for the classification of the chromosomes into seven groups is shown in the second row of Table 3. The first 31 genes in GA *chromosome* string represent 31 chromosome features and the last 4 genes indicate the number of hidden neurons. Since there are 15 “1” and 16 “0” in the first 31 genes, the optimized ANN-1 includes 15 features (input neurons). The selected 15 features correspond to features 1, 2, 3, 6, 8, 10, 12, 14, 15, 18, 19, 23, 28, 29, and 30. The last four genes (0110) represent that ANN-1 includes 6 hidden neurons. Using the independent validation dataset, ANN-1 correctly classifies and assigns 92.9% (2136 out of 2300) individual chromosomes into the correct sub-groups.

Table 4
The topologies of 7 ANNs used in the second layer of the classifier

ANN	2-1	2-2	2-3	2-4	2-5	2-6	2-7
Number of input neurons	13	14	14	17	15	14	11
Number of hidden neurons	14	15	5	13	14	10	8

In the second layer of the classifier, seven optimized ANNs are independently generated by the GA. Due to the different image characteristics of seven training and testing datasets as shown in Table 2, GA adaptively selects optimal topologies for these seven ANNs. The differences of the seven GA-optimized ANNs are summarized in Table 4. Each of the 31 initially computed image features are selected and used in at least once with the exception of feature 16, which corresponds to the total number of black bands in a chromosome and it has never been selected by any of seven ANNs. Also, no single feature has been selected and used in all seven ANNs. Fig. 8 shows the occurrence of each feature being used in the seven ANNs. Using the validation dataset to test the performance of these ANNs, the classification accuracy of the six ANNs, with the exception of the one used in sub-group C, ranges from 93.7% to 97.5%. The accuracy of the ANN optimized for sub-group C only reaches 67.5%. The overall classification accuracy is 86.8% and 86.7% for testing and validation datasets, respectively.

4. Discussion

This study focused on a task to improve the performance and robustness of an ANN-based automated karyotyping scheme by reducing the complexity of ANN structure and selecting optimal feature sets. The study has following unique characteristics. First, we developed and tested a multi-step classifier that includes two decision layers with eight ANNs. Since the classification task of each ANN was simplified by reducing the number of output neurons or classification classes, the sizes of these ANNs are substantially reduced compared with the sizes of the ANNs reported in previous studies. One important advantage of using our approach is that each ANN can be trained using relatively small dataset and keep highly robust performance. To prove this advantage, we also built and optimized a single ANN to classify all 24 types of chromosomes using the same GA optimization protocol. Then, the ANN was tested using the same validation dataset. This single ANN achieves 70% classification accuracy, while our two-layer classifier with eight ANNs achieves 86.7% classification accuracy.

Second, selection of an optimal set of low correlated features plays an important role in successfully developing ANN or other machine learning classifiers. The previous studies used empirical methods to select image features and the number of hidden neurons. In this study, we applied genetic algorithm (GA) to search for the optimal topology of the ANN including both the optimal feature set (input neurons) and the number of hidden neurons. Using GA optimization approach combined with the use of inde-

Table 3
Summary of GA optimization of 8 ANNs and classification results

Group name	Type of chromosomes	Best GA *chromosome*	Testing accuracy result (%)	Validation accuracy rate (%)
Seven groups	A–G	11100101010101100110001000011100110	91.1	92.9
A	1–3	11101000010000000001001111001111110	99.0	97.3
B	4–5	00010010110100100010111010111001111	94.0	95.5
C	6–12, X	01001001111010000001000111011010101	68.3	67.5
D	13–15	00000101000101101101111111010111101	95.0	95.3
E	16–18	01110110010100000011011110010011110	96.3	97.3
F	19–20	11110100001101100000010011000111010	96.5	98.0
G	21–22, Y	01100011010000000001101100110001000	94.7	93.7

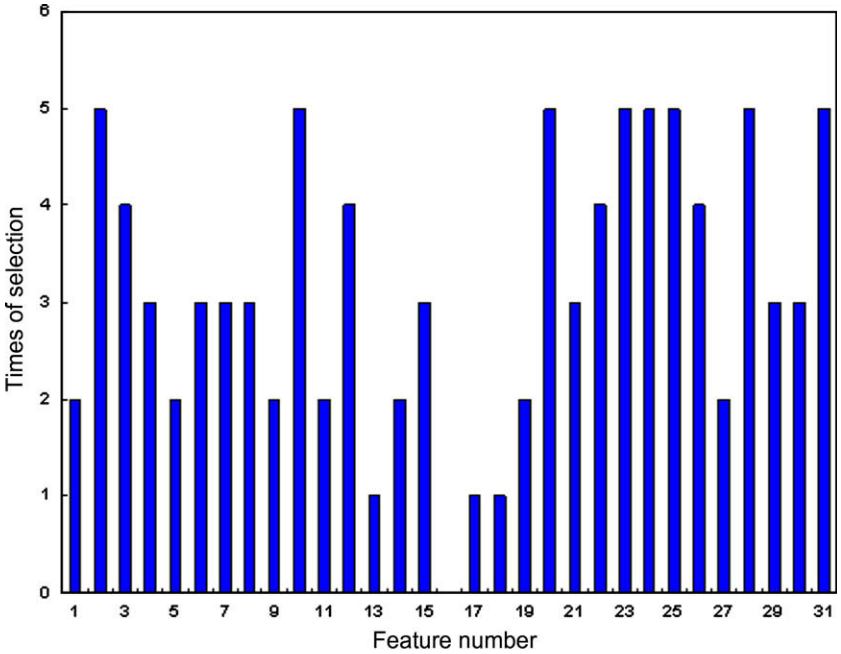

Fig. 8. Distribution of the feature selected in 7 ANNs.

pendent training and testing datasets, we substantially reduce the risk of the ANN being trapped by the local minimum in a multi-dimensional feature space. In addition, most of the previous studies [16,17,19] used a jackknifing (leave-one-out) training and testing method to optimize ANNs. This approach has inherent bias and does not generate a single workable ANN [41]. In order to eliminate or minimize the evaluation bias, we divided our database into three independent datasets and used the "training–testing–validation" method to optimize the classifier and assess its performance and robustness.

Third, unlike most of the previous studies that used publicly available datasets, we used the dataset collected from our own genetic laboratory. Table 5 describes and compares the image characteristics between our dataset and three public datasets (Copenhagen, Edinburgh, and Philadelphia). Since the Copenhagen database contains high quality straight chromosomes in which the locations of centromeres are known through manual identification, the testing performance of ANN classifiers is usually high. For example, three studies reported classification error rates of 6.2%, 8.8%, 10.3%, respectively, on Copenhagen database [9,16,17]. Edinburgh and Philadelphia datasets contain chromosomes that are much more difficult to classify. As a result, the performance of the previously developed classifiers is substantially low. The reported error rates are approximately 17.8% to 22.3% and 22.7% to 28.6% for Edinburgh and Philadelphia datasets, respectively [16,17,19]. One limitation of this study is that we cannot directly compare the performance of our scheme to other schemes due to the use of different databases. However, based on the characteristics presented in Table 5, the difficult level of our dataset should be between Edinburgh and Philadelphia databases. Thus, the 13.3% error rate of our classifier is encouraging and it suggests that our scheme achieves a very comparable or improved performance due to the diversity of our database and the use of an unbiased "training–testing–validation" method.

Fourth, we assessed robustness of our scheme using an independent validation dataset. The difference of classification accuracy of each ANN is small when applying to both the testing dataset used for GA optimization and the validation dataset. The biggest difference between applying to the two datasets is $\leqslant$1.7% in all sub-groups (as shown in Table 3). The high robustness of this new automated scheme can be attributed to two main factors. First, the classifier is composed of a number of adaptively optimized small size ANNs with a small number set of affective and low-correlated features. This adaptive approach eliminates the requirement of building and optimizing a large size ANN and the smaller size ANN is generally easy to be trained using relatively small training dataset and to maintain high level of robustness. For example, unlike some of previously developed ANNs that include more than 100 hidden neurons [16,17], the numbers of hidden neurons used in our eight ANNs are $\leqslant$17. The ANN used in the first layer has only 6 hidden neurons. Second, several measures have been taken to control or minimize the risk of over-fitting each ANN during the GA optimization. These measures include the selection of the large ratio between the training momentum and the learning rate, the limitation on ANN training iterations, and the use of the ANN performance on the testing dataset rather than the training dataset as the GA fitness criterion.

In summary, this preliminary study demonstrated (1) a new concept of using an adaptive optimization method to build an automated karyotyping scheme with an ANN-based two decision layer

Table 5
Summary of different chromosome databases

Database	Copenhagen	Edinburgh	Philadelphia	OUHSC
Tissue of origin	Peripheral blood	Peripheral blood	Chorionic villus	Peripheral blood amniotic fluid
Number of chromosomes	8106	5548	5847	6900
Data quality	Good	Fair	Poor	Fair
Including severely bent or touching chromosomes	No	Yes	Yes	Yes

classifier, (2) the feasibility of using GA to select optimal features and ANN topologies, and (3) the high and robust performance level of the new scheme. Despite of these encouraging results, this study has a number of weaknesses. First, the classification accuracy for sub-group C remains substantially lower compared to the other sub-groups of the chromosomes. Second, although we established a large and diverse database in this study, the metaphase chromosomes were all collected from normal samples. Because the numerical and structural changes can often be found in samples diagnosed with cancers and other genetic disorders [42], we do not know to date whether the performance of our scheme will be significantly affected when it is applied to classify abnormal chromosomes. Therefore, before this automated scheme can be introduced and applied in the routine clinical practice, we need to further improve the classification accuracy of the scheme (in particular for the sub-group C) and test the performance of our scheme in the classification of abnormal or cancerous metaphase chromosomes.

Acknowledgments

This research is supported in part by grants from the National Institutes of Health (NIH), CA115320. The authors acknowledge the support of the Charles and Jean Smith Chair endowment fund as well. The authors also thank Molly E. Donovan, for her editorial assistance.

References

[1] Tjio JH, Levan A. The chromosome number in man. Hereditas 1956;42:1–6.

[2] Conference D. A proposed standard system of nomenclature of human mitotic chromosomes. Lancet 1960;1:1063–5.

[3] Piper J, Granum E, Rutovitz D, Ruttledge H. Automation of chromosome analysis. Signal Process 1980;2:203–21.

[4] Wang X, Zheng B, Wood M, Li S, Chen W, Liu H. Development and evaluation of automated systems for detection and classification of banded chromosomes: current status and future perspectives. J Phys D: Appl Phys 2005;38:2536–42.

[5] Groen F, Kate Tt, Smeulders A, Young I. Human chromosome classification based on local band descriptors. Patt Recogn Lett 1989;9:211–22.

[6] Piper J, Granum E. On fully automatic feature measurement for banded chromosome classification. Cytometry 1989;10(3):242–55.

[7] van Vliet LJ, Young IT, Mayall BH. The Athena semi-automated karyotyping system. Cytometry 1990:51–8.

[8] Schwartzkopf WC. Maximum likelihood techniques for joint segmentation-classification of multi-spectral chromosome images. Austin: The University of Texas at Austin; 2002.

[9] Jennings AM, Graham J. A neural network approach to automatic chromosome classification. Phys Med Biol 1993;38:959–70.

[10] Wu Q, Castleman KR. Automated chromosome classification using wavelet-based band pattern descriptors. IEEE Symposium on Computer-Based Medical Systems CBMS 2000. Houston, TX, USA: 2000. p. 189–94.

[11] Johnston DA, Tang KS, Zimmerman S. Band features as classification measures for G-banded chromosome analysis. Comput Biol Med 1993;23(2):115–29.

[12] Tso MKS, Graham J. The transportation algorithm as an aid to chromosome classification. Patt Recog Lett 1983;1:489–96.

[13] Lerner B, Levinstein M, Rosenberg B, Guterman H. Feature selection and chromosome classification using a multilayer perceptron neural network, Neural Networks. IEEE Int Conf Comput Intell 1994:3540–5.

[14] Wu Q, Liu Z, Chen T, Xiong ZKRC. Subspace-based prototyping and classification of chromosome images. IEEE Trans Image Process 2005;14:1277–87.

[15] Carothers A, Piper J. Computer-aided classification of human chromosomes: a review. Stat Comput 1994;4:161–71.

[16] Sweeney WP, Musavi MT, J.N.Guidi. Classification of chromosomes using a probabilistic neural network. Cytometry 1996;16:17–24.

[17] Errington P, Graham J. Application of artificial neural networks to chromosome classification. Cytometry 1993;14:627–39.

[18] Lerner B. Toward a completely automatic neural-network-based human chromosome analysis. IEEE Trans Syst Man Cybern—B: Cybern 1998;28: 544–52.

[19] Cho J. Chromosome classification using back propagation neural networks. IEEE Eng Med Biol Mag 2000;19:28–33.

[20] Delshadpour S. Reduced size multi layer perceptron neural network for human chromosome classification. In: Proceedings of the 25th Annual International Conference of the IEEE (Engineering in Medicine and Biology Society). 2003. p. 2249–52.

[21] Wu Q, Suetens P, Oosterlinck A. Chromosome classification using a multi-layer perceptron neural net. In: Proceedings of the 12th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 1990. p. 1453–54.

[22] Wu Q, Suetens P, Oosterlinck A. On knowledge-based improvement of biomedical pattern recognition-a case study. In: Proceedings of 5th conference on artificial intelligence for applications. 1989. p. 239–44.

[23] Lu Y, Ya Y. An expert system for banded chromosomes recognition. In: Proceedings of the annual international conference of the IEEE engineering in medicine and biology society. 1989. p. 1789–90.

[24] Ramstein G, Bernadet M, Kangoud A, Barba D. A rule-based image analysis system for chromosome classification. In: Proceedings of the annual international conference of the IEEE engineering in medicine and biology society; 1992; 1992. p. 926–27.

[25] Zimmerman SO, Johnston DA, Arrighi FE, Rupp ME. Automated homologue matching of human G-banded chromosomes. Comput Biol Med 1986;16:223–33.

[26] Keller JM, Gader P, Sjahputera O, Caldwell CW. A fuzzy logic rule-based system for chromosome recognition. In: Proceedings of the eighth IEEE symposium on computer-based medical systems. 1995. p. 125–32.

[27] Popescu M, Gader P, Keller J, Klein C. Automatic karyotyping of metaphase cells with overlapping chromosomes. Comput Biol Med 1999;29:61–82.

[28] Stanley RJ, Keller JM, Gader P, Caldwell CW. Data-driven homologue matching for chromosome identification. IEEE Trans Med Imaging 1998;17:451–62.

[29] Gregor J, Granum E. Finding chromosome centromeres using band pattern information. Comput Biol Med 1991;21(1-2):55–67.

[30] Graham J, Errington P, Jennings AM. A neural network chromosome classifier. J Radiat Res 1992;33:250–7.

[31] Mitchell TM. Machine learning. Boston MA: WCB McGraw-Hill; 1997.

[32] Li Q. Reliable evaluation of performance level for computer-aided diagnostic scheme. Acad Radiol 2007;14:985–91.

[33] Wang X, Li S, Liu H, Wood M, Chen W, Zheng B. Automated identification of analyzable metaphase chromosomes depicted on microscopic digital images. J Biomed Inform 2007;41:264–71.

[34] Wang X, Zheng B, Li S, Mulvihill JJ, Liu H. A rule-based scheme for centromere identification and polarity assignment of metaphase chromosomes. Comput Methods Programs Biomed 2008;89(1):33–42.

[35] Kramer HP, Bruckner JB. Iterations of a non-linear transformation for enhancement of digital images. Patt Recogn 1975;7:53–8.

[36] Granum E. Pattern recognition aspects of chromosome analysis—computerized and visual interpretation of banded human chromosomes. Lyngby: Tech Univ Denmark; 1980.

[37] Tseng CC. Human chromosome analysis in tested studies for laboratory teaching. In: Goldman CA, editor. In: Proceedings of the 16th Workshop/ Conference of the Association for Biology Laboratory Education (ABLE). Atlanta, Georgia; 1995. p. 35–56.

[38] Holland JH. Adaption in neural and artificial systems. University of Michigan press. Ann Arbor Mich 1975.

[39] Zheng B, Chang YH, Good WF, Gur D. Performance gain in computer-assisted detection schemes by averaging scores generated from artificial neural networks with adaptive filtering. Med Phys 2001;28:2302–8.

[40] Kantrowitz M. Prime time freeware for AI, issue 1-1. Artif Intell Repository 1994;1(1). Selected materials from the Carnegie Mellon University, Sunnyvale, CA, Prime Time Freeware.

[41] Li Q, Doi K. Reduction of bias and variance for evaluation of compute-aided diagnostic schemes. Med Phys 2006;33:868–75.

[42] Shih LM, Wang TL. Apply innovative technologies to explore cancer genome. Curr Opin Oncol 2005;17:33–8.