0023-6837/00/8011-1629$03.00/0
LABORATORY INVESTIGATION

Vol. 80, No. 11, p. 1629, 2000
Printed in U.S.A.

Chromosome Identification Using Hidden Markov Models: Comparison with Neural Networks, Singular Value Decomposition, Principal Components Analysis, and Fisher Discriminant Analysis

John M. Conroy, Tamara G. Kolda, Dianne P. O'Leary, and Timothy J. O'Leary

Center for Computing Sciences (JMC), Institute for Defense Analyses, Bowie, Maryland; and Computational Sciences and Mathematics Research Department (TGK), Sandia National Laboratories, Livermore, California; and Computer Science Department and Institute for Advanced Computer Studies (DPO), University of Maryland, College Park, Maryland; and Department of Cellular Pathology (TJO), Armed Forces Institute of Pathology, Washington, DC

SUMMARY: The analysis of G-banded chromosomes remains the most important tool available to the clinical cytogeneticist. The analysis is laborious when performed manually, and the utility of automated chromosome identification algorithms has been limited by the fact that classification accuracy of these methods seldom exceeds about 80% in routine practice. In this study, we use four new approaches to automated chromosome identification — singular value decomposition (SVD), principal components analysis (PCA), Fisher discriminant analysis (FDA), and hidden Markov models (HMM) — to classify three well-known chromosome data sets (Philadelphia, Edinburgh, and Copenhagen), comparing these approaches with the use of neural networks (NN). We show that the HMM is a particularly robust approach to identification that attains classification accuracies of up to 97% for normal chromosomes and retains classification accuracies of up to 95% when chromosome telomeres are truncated or small portions of the chromosome are inverted. This represents a substantial improvement of the classification accuracy for normal chromosomes, and a doubling in classification accuracy for truncated chromosomes and those with inversions, as compared with NN-based methods. HMMs thus appear to be a promising approach for the automated identification of both normal and abnormal G-banded chromosomes. (*Lab Invest 2000, 80:1629–1641*).

Although the use of spectral karyotyping (Macville et al, 1997; Schrock et al, 1997; Veldman et al, 1997) is redefining the role of G-banding in chromosome analysis, analysis of chromosome banding patterns remains a cornerstone of karyotypic analysis both for routine diagnosis and for application in such techniques as comparative genomic hybridization (Piper et al, 1995). Chromosome classification and analysis is aided by the use of automated karyotyping systems that yield a preliminary classification for each chromosome, which may be corrected by hand as necessary. Automated karyotyping relies upon acquisition of a digital image, followed by extraction of chromosome features. Two general approaches to feature extraction are employed: gray level encoding of each chromosome and more complex extraction of distinctive features. These features may then be used in an algorithm that assigns the chromosome to one of 24 classes (autosomes 1–22, X, and Y). A variety of such algorithms has been proposed, based upon approaches such as Bayesian analysis (Lundsteen et al, 1986), Markov networks (Granum and Thomason, 1990; Guthrie et al, 1993), neural networks (NN) (Beksac et al, 1996; Errington and Graham, 1993; Graham et al, 1992; Jennings and Graham, 1993; Korning, 1995; Leon et al, 1996; Malet et al, 1992; Sweeney et al, 1994; Sweeney et al, 1997), and simple feature matching (Piper and Granum, 1989). The reported classification accuracy varies surprisingly little by approach. Most methods achieve approximately 90% correct classification of the Copenhagen chromosome data set; commercial implementations typically achieve approximately 80% correct classification in routine use.

Automated chromosome classification entails several steps. First, an image segmentation step is used to create distinct images of each chromosome in a metaphase. Then, salient features of the chromosome image are extracted. Typically, gray level encoding is employed to represent the chromosome by a vector of

Received May 16, 2000.
The work of Dianne O'Leary was supported by NSF Grant CCR 97-32022. The work of Tamara Kolda was supported by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy, under contracts DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation, and DE-AC04-94AL85000 with Sandia Corporation.
Address reprint requests to: Timothy J. O'Leary, Department of Cellular Pathology, Armed Forces Institute of Pathology, 14th Street and Alaska Avenue, NW, Washington, DC 20306-6000. Fax: 202 782 7623; E-mail: oleary@afip.osd.mil

gray level values, which are obtained by sampling at evenly spaced intervals along the chromosome's medial axis. (See, for example, Errington and Graham, 1993.) Different vectors may contain a different number of samples, so vectors are typically stretched or compressed to a fixed number of entries via constant interpolation or downsampling. Because variations in lighting can cause the gray scale measurements to vary, all stretched vectors are normalized to Euclidean magnitude 1. Figure 1 illustrates this stretching, and Figure 2 illustrates the variations in measured values for chromosomes having the same identity. Chromosome 1 is usually the easiest to identify; it is physically the longest chromosome, and the banding pattern is particularly distinctive. The Y chromosome is among the hardest; it is physically relatively short, and the banding pattern is often rather indistinct.

Feature extraction provides an alternative to gray level encoding. Piper and Granum (1989), for example, have proposed the use of 30 classification parameters derived from automated measurements. These features include the following:

- physical length of the chromosome
- location of the centromere (a narrowed region of the chromosome)
- the area of the chromosome
- the perimeter of the convex hull of the chromosome
- the number of bands
- inner products of the gray level values with various basis vectors resembling a set of wavelet "hat" functions.

In summary, the problem is to assign an identity (1–22, X, or Y) to a chromosome, given a vector containing its gray level measurements or other measured features and some training vectors with known identities. Helpful additional output would include degree of certainty in identification, identification of abnormal chromosomes, and automatic characterization of abnormalities.

In this paper we propose some new approaches for solving the problem of automated chromosome identification:

- singular value decomposition
- principal component analysis
- Fisher discriminant analysis
- hidden Markov models.

A brief description of each approach follows; more details may be found in the Appendix.

Singular Value Decomposition (SVD)

One way to pose our problem is to seek among all vectors in the training set the one that most closely matches the vector of unknown identity. We then assign the unknown vector the identity of this most closely matching chromosome. Viewed in this way, the problem resembles the retrieval of a document whose keywords most closely match those of a query. We represent each document and query by a vector indicating the relative importance of each keyword. Literal matching (eg, taking inner products of document vectors with the query and then choosing the maximum) is not usually the best strategy because latent relationships and document clusters are not revealed.

Instead, in the latent semantic indexing (LSI) method (Berry et al, 1995; Deerwester et al, 1990), the vectors characterizing the documents form the columns of the document matrix. We approximate this matrix by a low-rank matrix, and then scoring is done by inner products of the query with the low-rank approximation.

Principal Component Analysis (PCA)

The SVD algorithm implicitly assumes that the measurements are independent and have similar standard deviations. If we wish to take covariances into account, then we need to use the SVD in a somewhat different way. Rather than finding the identity with the largest score, we find the one with the minimal Mahalanobis distance to the mean of the training chromosomes of that identity.

Fisher Discriminant Analysis (FDA)

Fisher discriminant analysis (Mardia et al, 1979) is similar to principal component analysis in that it uses a multidimensional normal distribution to model the 24 clusters. Also, like PCA, it projects the data into a

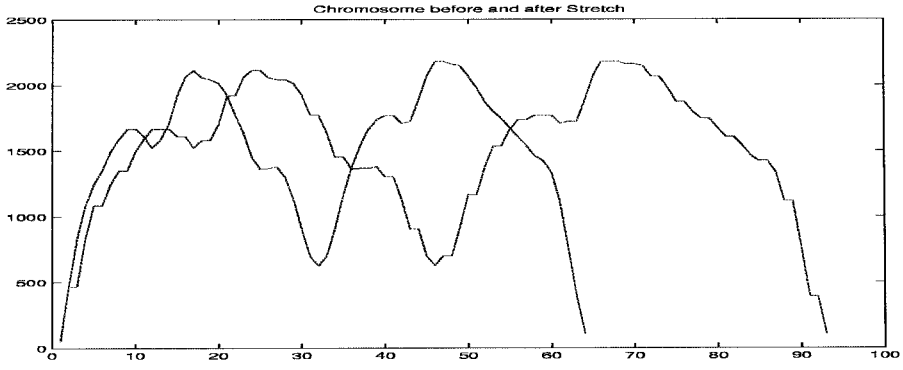

Figure 1.

Stretching a chromosome. The ordinate (y-axis) shows the gray level (staining intensity) as a function of the position along the chromosome, shown on the x-axis (from the p-terminus on the left to the q-terminus on the right). The chromosome has been stretched from 64 pixels to 93 pixels in length.

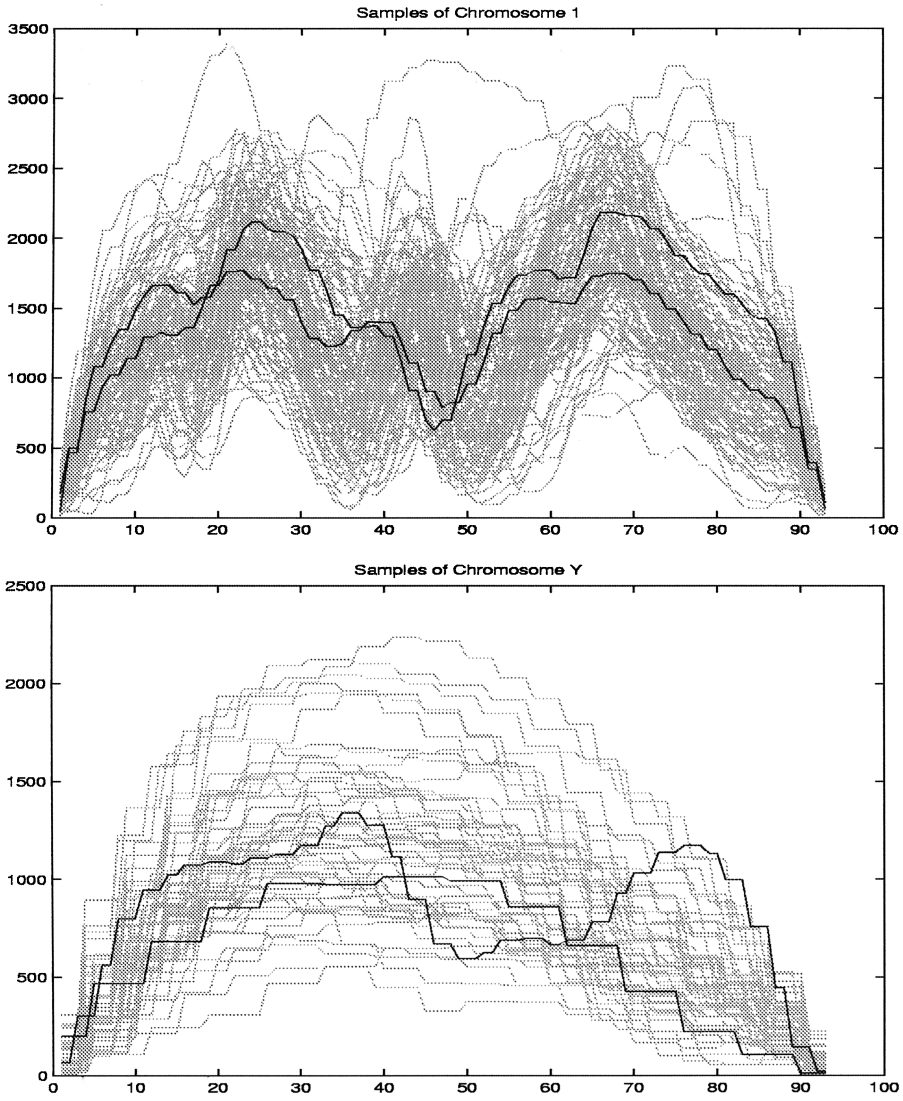

Figure 2.

Samples of chromosome 1 and the Y chromosome from the Edinburgh data set, with two samples of each highlighted for clarity. The ordinate (y-axis) shows the gray level (staining intensity) as a function of the position along the chromosome, shown on the x-axis axis (from the p-terminus on the left to the q-terminus on the right).

lower dimensional space. This projection is not done via the SVD but rather by solving a generalized eigenvalue problem. The projection is computed using the training data so as to maximize the ratio of the *between cluster* distances to the *within cluster* distances.

Hidden Markov Models (HMM)

One characteristic of speech problems as well as chromosome karyotyping is that the vectors can be of variable length. For instance, the duration of sound for a given phrase varies from speaker to speaker and even from trial to trial. Similarly, the number of gray levels sampled from a chromosome is variable. The SVD, PCA, and NN models all must normalize the input vector to a fixed number of entries, but hidden Markov models (HMM) (Baum and Eagon, 1967; Baum et al, 1970) have no such restrictions. (See Rabiner, 1989, and Rabiner and Juang, 1986, for an introduction to these methods.)

The models that we build work with a sequence of gray level "triples" (Fig. 3). From the vector of gray scale observations, we form a vector of first differences and a vector of second differences. Our 24 models output triples of observations approximating those of typical chromosomes of each identity. An observer, then, would see a sequence of gray level triples, each representing a single entry from each of the three vectors. Hidden from the observer is a Markov chain that is generating the output. The current state of the chain produces a single output triple, and then a new state is chosen according to probabilities specified in a transition probability matrix. For each sequence of gray level triples, we compute a probability that the sequence was generated by each of the 24 models. This 24-element vector of scores is used to classify each unknown chromosome. The details of

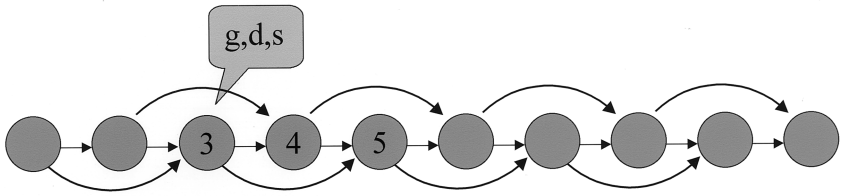

Figure 3.
Graphic illustration of a hidden Markov model with 10 states. If the model is currently in state 3, then it outputs a triple of values (gray level, first difference, and second difference) chosen according to the output function *B*. Then, with probability a_{next} the model transitions to state 4, with probability a_{skip} it transitions to state 5, and otherwise it stays in state 3 for another cycle.

the HMM classifier are given in the Appendix of this article.

Results

Experiment 1

First we compare our results with those of the neural net model of Errington and Graham (1993). For each of the three data sets (Philadelphia, Edinburgh, and Copenhagen), we display in Table 1 the percentage of chromosomes classified correctly by each model. We note that HMM frequently gives the best performance. SVD performs well and is generally better than PCA. The second differences have little effect on the performance of the neural nets although the first differences improve performance slightly.

Experiment 2

In this experiment, we explore the robustness of the methods when there are "mild" chromosomal abnormalities present. We degrade the data from each scoring chromosome (but not the training chromosomes) by taking the sequence of gray level values in the middle 10% and reversing their order. This simulates an internal inversion of chromosomal material.

The behavior of the best methods from Experiment 1 are shown in Table 2. The HMM performs the best, degrading by at most 8 percentage points. The other three methods do not behave as well, but each achieves at least 56% accuracy.

Experiment 3

Next we degrade the data by truncating each of the scoring chromosomes (but not the training chromosomes) by deleting either the first or last 10% of the gray level values in each sequence. This simulates an artifact commonly encountered during the "editing phase" of semiautomated karyotype analysis, in which overlapping chromosomes are "cut apart," in addition to those deletions of the terminal chromosome arms that occur "naturally." In Table 3 we see that the HMM is quite robust on this data, degrading by, at most, 4 percentage points. The SVD is moderately successful, but PCA and NN methods classify most chromosomes incorrectly.

Experiment 4

We also tested several algorithms on the feature data in an experiment analogous to that of Errington and Graham (1993). From the data in Table 4, we conclude that the best methods were PCA and FDA, both of

Table 1. Results of Experiment 1

Gray level data alone	Phi	Edi	Cph
		%	
HMM, no differences	59	67	84
HMM, 1 difference	70	75	91
HMM, 2 differences	[a]73	[a]78	92
Rank 24 SVD, no differences	65	71	91
Rank 24 SVD, 1 difference	71	[a]78	[a]93
Rank 24 SVD, 2 differences	71	[a]78	[a]93
PCA, no differences	64	72	86
PCA, 1 difference	68	76	86
PCA, 2 differences	68	75	86
NN, no differences	69	72	90
NN, 1 difference	69	74	92
NN, 2 differences	70	73	92
Errington and Graham (no differences)	71	[a]78	91

Phi, Philadelphia; Edi, Edinburgh; Cph, Copenhagen; HMM, hidden Markov model; SVD, singular value decomposition; PCA, principal components analysis; NN, neural network.
[a] Best performances.

Table 2. Results of Experiment 2

Internal Inversion	Phi	Edi	Cph
		%	
HMM, 2 differences	[a]66	[a]70	[a]86
Rank 24 SVD, 2 differences	60	63	71
PCA, 2 differences	56	65	76
NN, 1 difference[b]	57	60	77

[a] Best performances.
[b] In this experiment, NN with 1 difference performed slightly better than NN with 2 differences.

Table 3. Results of Experiment 3

Truncated sequences	Phi	Edi %	Cph
HMM, 2 differences	[a]70	[a]74	[a]89
Rank 24 SVD, 2 differences	52	55	48
PCA, 2 differences	36	46	32
NN, 1 difference[b]	35	43	45

[a] Best performances.
[b] In this experiment, NN with 1 difference performed slightly better than NN with 2 differences.

Table 4. Results of Experiment 4

Feature data alone	Phi	Edi %	Cph
Rank 24 SVD	73	80	94
PCA	82	85	[a]95
FDA	[a]83	[a]86	[a]95
Errington and Graham	77	82	94

FDA, Fisher discriminant analysis.
[a] Best performances.

which performed slightly better than the NN of Errington and Graham (1993).

Experiment 5

Under the assumption that each metaphase consists of chromosomes from a single cell, classification errors can be further reduced by adding the constraint that the slide produced for a single cell contains, at most, two copies of the autosomes, and either 2 X's or one X and one Y. This assumption is valid in some special cases, such as chromosome spreads produced for comparative genomic hybridization. To illustrate how this information can be used, we consider the results of PCA for the feature data. Given the FDA scores we can form two likelihood matrices, F and M, where F corresponds to the assumption that the patient is female and M that the patient is male. The likelihood matrix F is formed based on the FDA log likelihoods as specified by the following equation:

$$F_{ij} = \begin{cases} \text{likelihood } i \text{ is type } j/2 \\ \qquad \text{for } j = 1, \ldots, 44 \text{ and for } i = 1, \ldots, 46 \\ \text{likelihood } i \text{ is type X} \\ \qquad \text{for } j = 45, 46 \text{ and for } i = 1, \ldots, 46 \end{cases}$$

The matrix M is defined analogously with columns 45 and 46 corresponding to the likelihoods of chromosome X and Y respectively. The total likelihood of assigning labels to the chromosomes is maximized by solving two linear programs of a special type, a linear assignment problem. The linear assignment problem finds a matching of the rows to the columns with maximum sum. There are a number of very efficient polynomial methods for solving this problem (Papadimitriou and Steiglitz, 1982). The method used here was the Hungarian algorithm, the work of which is proportional to the cube of the number of rows. Both the Copenhagen and Edinburgh data sets have the property that each metaphase consists of chromosomes from a single cell. (A number of the metaphases from the Philadelphia data set had 47 chromosomes identified on them.) Tables 5 and 6 give the results of this linear assignment given gray scale values or feature vectors. For the Edinburgh and Copenhagen data sets, linear assignment on the results of the HMM for gray levels with 2 differences improved the accuracy by 4 to 6 percentage points. It improved the accuracy by 6 to 8 percentage points for the truncated sequences and by 9 percentage points on the chromosomes in which the centers were inverted. For feature data, the results improved by 2 or 3 percentage points and were slightly better than those achieved by Sweeney et al (1994).

Discussion

Interpretation of G-banded chromosomes remains the cornerstone of both routine karyotyping and chromosome identification for such molecular biologic methods as comparative genomic hybridization. Although algorithms intended to speed up karyotypic analysis are widely available, their use has been limited by their modest accuracy in classifying even "normal" chromosomes. Methods that are sufficiently robust to accurately classify chromosomes that are abnormal, as a result of disease, constitutional anomaly, or artifacts introduced during acquisition of chromosome images, have not been previously published.

In this paper, we demonstrate that although a number of algorithms achieve 90% accurate classification of "normal" chromosomes, most of these algorithms perform poorly when as little as 10% at the end of a chromosome arm is truncated. This amount of truncation is commonly encountered in practice. Although it usually results from artifacts associated with the acquisition and processing of digital data, truncation may also be characteristic of disease. Although the performance of these algorithms is better for chromosomes in which a small internal inversion has been simulated, the rate of correct classification is reduced by 6% to 22% even for these chromosomes. Automated classification methods are thus significantly less useful in routine practice than the putative 90%

Table 5. Results of Experiment 5: Gray Level Data

Gray level data alone	Edi %	Cph
HMM	78	92
HMM with linear assignment	84	96
Inversion		
HMM	70	86
HMM with linear assignment	79	95
Truncated Sequences		
HMM	74	89
HMM with linear assignment	82	95

Table 6. Results of Experiment 5: Feature Data

Feature data alone	Edi	Cph
Feature data	86	95
Feature data with linear assignment	89	97
Sweeney, Musavi, and Guidi with constraint	84	96

classification accuracy would suggest. Our results show that one type of classification algorithm, HMM, is significantly better at correctly identifying chromosomes bearing these abnormalities than are either the other novel algorithms we explored or the neural network algorithms in common use. Introducing a commonly encountered artifact — truncation of the terminal 10% of either the p- or q-arm, resulted in a 5% to 11% degradation in classification accuracy. In contrast, the next best method (Rank 24 SVD with 2 differences) had a 22% reduction in accuracy on truncated chromosomes from the Copenhagen data set. Our results further demonstrate the utility of constrained classification algorithms that rely on the observation that normal cells carry, at most, 2 of each autosome, and either 2 X chromosomes, or an X and a Y. These algorithms achieve classification accuracies for normal chromosomes of 95% to 97% (Copenhagen data set), even for truncated sequences. This represents a reduction of approximately 50% in the classification error rate. Although this constraint is inappropriate in cases where cytogenetic anomaly is being sought (such as prenatal diagnosis or cytogenetic characterization of tumor specimens), it is useful and appropriate in applications such as comparative genomic hybridization, in which metaphase spreads are prepared from cell cultures of "normal" individuals.

HMMs have previously proven useful in several areas of biological science, including speech recognition (Jelinek, 1995), EKG analysis (Koski, 1996), gene identification (Lukashin and Borodovsky, 1998), and protein structure prediction (Sonnhammer et al, 1998). These problems are similar in that all involve classification of data sequences (vectors) that can demonstrate substantial within-class variations in length and pattern. HMMs appear to be especially well-suited to solving such problems, because the classification resulting from the model does not depend upon the precise location of values within the data vector, but rather upon the relationships between adjacent or nearly adjacent data values. This feature of HMMs is very useful in chromosome classification, because the same chromosome (chromosome 5, for example) can vary substantially in length among various metaphase spreads. This feature alone gives HMMs a robustness that is not found in most other classification approaches. One result is that chromosome classification using HMMs created using normal chromosomes is expected to remain reliable for substantially larger truncations/terminal deletions and internal inversions than were explored in this paper.

HMMs are also expected to be useful in the characterization of abnormal karyotypes for which training data is not available. For example, one can synthetically create HMMs characteristic of reciprocal t(14; 18) translocations with varying break points, based upon data obtained from normal chromosomes 14 and 18. (This can also be done with NN, SVD, PCA, and FDA.) By competitively scoring these models, we may expect to obtain a fairly precise localization of the break point if a chromosome bearing t(14; 18) is encountered in a test set. By creating such "synthetic" HMMs for chromosomes bearing truncations and deletions, we may expect to further improve the classification of chromosomes bearing these anomalies as well.

In summary, we have applied four mathematical approaches for automated chromosome identification: singular value decomposition (SVD), principal components analysis (PCA), Fisher discriminant analysis (FDA), and hidden Markov models (HMM). We have demonstrated that although all these approaches yield similar results for "perfect" normal chromosomes, the HMM approach is superior for the identification of imperfect and/or abnormal chromosomes. Finally, we expect that the HMM approach can be implemented in a way that allows highly accurate classifications to be made even when few data are available upon which to train new models.

Materials and Methods

Data Preparation

The Copenhagen, Edinburgh, and Philadelphia data sets were used in creating and validating the mathematical models. These data sets consist of vectors of the gray values obtained from linear axial traces of 5100 to 8100 chromosomes each, together with the chromosome assignment. Each of these data sets was divided into "training" and "scoring" parts as done by Errington and Graham (1993).

Classification Experiments

A more precise description of each of the mathematical models underlying our classification experiments is given in the Appendix. The data for our SVD, PCA, and FDA experiments were the gray level vectors, augmented by their first and second differences. We computed differences between vector elements, and then stretched them to equal lengths (the length of the longest training vector). The first and second differences were weighted by a factor of 5. These vectors were then used without further preprocessing.

For experiments, we set the rank of the SVD approximations to $K = 24$. A rank of 36 improves the results slightly but does not seem to be worth the extra computational effort.

Back-propagation networks were created and run using Netmaker Professional for Windows and Brainmaker Professional for Windows (California Scientific Software, Nevada City, California). Networks consisted of an input layer of 15, 30, or 45 nodes, where only gray levels, gray levels plus first differences, or gray levels plus first and second differences were used as network input. A single hidden layer of 200 nodes was used. Network output was a 24-element

vector, in which each element represented one chromosome type. In the training phase, each chromosome was represented by both the 15-, 30-, or 45-element input vector and a 24-element classification vector in which all elements were set to 0 except for that element corresponding to the encoding chromosome. Training was accomplished using a constant "learning rate" of 0.05, with a training tolerance of 0.4. When 75% correct classification of the training set was achieved, the training tolerance was reduced by a factor of 0.9. Training was discontinued after 1000 iterations in which each chromosome in the training set was presented to the network. During the testing phase, a chromosome was considered to be correctly identified by the neural network if the largest element in the neural network output vector corresponded to the correct chromosome number.

For the HMM, the training data was further subdivided by chromosome types. A HMM was then found for each chromosome type, as discussed in the Appendix. The number of states was set to be the median length of chromosomes in the training data. [Au: Anonymous, 1997; Golub and Van Loan, 1996; Gu and Eisenstat, 1993; Jackson, 1991; Rabiner and Juang, 1993 have not been cited in the text. Please cite or remove from reference list.]

Acknowledgements

We are grateful to Robert L. Becker for valuable discussions and to Mohamad T. Musavi for providing the chromosome data sets and helping with their use.

References

Baum LE and Eagon JA (1967). An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. Bull Am Math Soc 73:360–363.

Baum LE, Petrie T, Soules G, and Weiss N (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41:164–171.

Beksac MS, Eskiizmirliler S, Cakar AN, Erkmen AM, Dagdeviren A, and Lundsteen C (1996). An expert diagnostic system based on neural networks and image analysis techniques in the field of automated cytogenetics. Technol Health Care 3:217–229.

Berry MW, Dumais ST, and O'Brien GW (1995). Using linear algebra for intelligent information retrieval. SIAM Review 37:573–595.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, and Harshman R (1990). Indexing by latent semantic analysis. J Soc Inform Sci 41:391–407.

Errington PA and Graham J (1993). Application of artificial neural networks to chromosome classification. Cytometry 14:627–639.

Graham J, Errington P, and Jennings A (1992). A neural network chromosome classifier. J Radiat Res (Tokyo) 33(Suppl):250–257.

Granum E and Thomason MG (1990). Automatically inferred Markov network models for classification of chromosomal band pattern structures. Cytometry 11:26–39.

Guthrie C, Gregor J, and Thomason MG (1993). Constrained Markov networks for automated analysis of G-banded chromosomes. Comput Biol Med 23:105–114.

Jelinek F (1995). Training and search methods for speech recognition. Proc Natl Acad Sci USA 92:9964–9969.

Jennings AM and Graham J (1993). A neural network approach to automatic chromosome classification. Phys Med Biol 38:959–970.

Korning PG (1995). Training neural networks by means of genetic algorithms working on very long chromosomes. Int J Neural Syst 6:299–316.

Koski A (1996). Modelling ECG signals with hidden Markov models. Artif Intell Med 8:453–471.

Leon MA, Gader P, and Keller J (1996). Multiple neural network response variability as a predictor of neural network accuracy for chromosome recognition. Biomed Sci Instrum 32:31–37.

Lukashin AV and Borodovsky M (1998). GeneMark.hmm: new solutions for gene finding. Nucleic Acids Res 26:1107–1115.

Lundsteen C, Gerdes T, and Maahr J (1986). Automated classification of chromosomes as part of a routine system for clinical analysis. Cytometry 7:1–7.

Macville M, Veldman T, Padilla-Nash H, Wangsa D, O'Brien P, Schrock E, and Ried T (1997). Spectral karyotyping, a 24-colour FISH technique for the identification of chromosomal rearrangements. Histochem Cell Biol 108:299–305.

Malet P, Benkhalifa M, Perissel B, Geneix A, and Le Corvaisier B (1992). Chromosome analysis by image processing in a computerized environment. Clinical applications. J Radiat Res (Tokyo) 33(Suppl):171–188.

Mardia KV, Kent JT, and Bibby JM (1979). Multivariate analysis. New York: Academic Press.

Papadimitriou CH and Steiglitz K (1982). Combinatorial optimization: Algorithms and complexity. Englewood Cliffs, NJ: Prentice Hall, 247–255.

Piper J and Granum E (1989). On fully automated feature measurement for banded chromosome classification. Cytometry 10:242–255.

Piper J, Rutovitz D, Sudar D, Kallioniemi A, Kallioniemi OP, Waldman FM, Gray JW, and Pinsky S (1995). Computer image analysis of comparative genomic hybridization. Cytometry 19:10–26.

Rabiner LR (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77: 257–285.

Rabiner LR and Juang BH (1986). An introduction to hidden Markov models. IEEE ASSP Magazine January:4–16.

Schrock E, Veldman T, Padilla-Nash H, Ning Y, Spurbeck J, Jalal S, Shaffer LG, Papenhausen P, Kozma C, Phelan MC, Kjeldsen E, Schonberg SA, O'Brien P, Biesecker L, du Manoir S, and Ried T (1997). Spectral karyotyping refines cytogenetic diagnostics of constitutional chromosomal abnormalities. Hum Genet 101:255–262.

Sonnhammer EL, von Heijne G, and Krogh A (1998). A hidden Markov model for predicting transmembrane helices in protein sequences. Ismb 6:175–182.

Sweeney N, Becker RL, and Sweeney B (1997). A comparison of wavelet and Fourier descriptors for a neural network chromosome classifier. Proceedings of the IEEE Conference on Engineering in Medicine and Biology. Chicago: IEEE Publications.

Sweeney WPJ, Musavi MT, and Guidi JN (1994). Classification of chromosomes using a probabilistic neural network. Cytometry 16:17–24.

Veldman T, Vignon C, Schrock E, Rowley JD, and Ried T (1997). Hidden chromosome abnormalities in haematological malignancies detected by multicolour spectral karyotyping. Nat Genet 15:406–410.

Appendix

Preparing the Data

In order to use the SVD or PCA methods, all chromosome vectors must have the same number of entries. We discuss in this section how we create these vectors.

Suppose that we have recorded ℓ gray level values for a chromosome: $z_1, \ldots, z_\ell$, and let $\hat{\ell}$ be the length to which we stretch all chromosomes. We create a stretched vector x of length $\hat{\ell}$, where the jth entry in x is equal to the ith entry in z, with i equal to

$$\frac{\ell - 1}{\hat{\ell} - 1}(j - 1) + 1$$

rounded to the nearest integer between 1 and ℓ.

We sometimes use *first differences* and *second differences* for added information. The first and second differences of z are denoted z' and z'', respectively, and defined as

$$z' = \begin{bmatrix} z_2 - z_1 \\ (z_3 - z_1)/2 \\ \vdots \\ (z_\ell - z_{\ell-2})/2 \\ z_\ell - z_{\ell-1} \end{bmatrix}$$

$$\text{and } z'' = \begin{bmatrix} (2z_1 - 3z_2 + 4z_3 - z_4)/4 \\ (z_1 - 2z_2 + z_3)/4 \\ \vdots \\ (z_{\ell-2} - 2z_{\ell-1} + z_\ell)/4 \\ (-z_{\ell-3} + 4z_{\ell-2} - 5z_{\ell-1} + 2z_\ell)/4 \end{bmatrix}.$$

We then stretch z' and z'' in the same way we stretched z, creating x' and x''.

Each chromosome vector x, perhaps with x' and x'' appended to it, becomes a column in the matrix X used for training. There are M columns in this matrix, where M is the number of training chromosomes, and the number of rows L is either $\hat{\ell}$, $2\hat{\ell}$, or $3\hat{\ell}$, depending on whether differences are being used.

If we are using "feature" data, then we simply let the $L \times M$ matrix X denote the matrix of features where L denotes the number of features and each column is the data for a different chromosome.

SVD Method

One way to pose our problem is to seek the chromosome in the training set that most closely matches the chromosome of unknown type. Viewed in this way, the problem resembles textual information retrieval where the goal is to find a document that most closely matches a given query. Each document and query is represented by a vector of keyword weights (e.g., the keyword "dog" might be assigned a weight of 3 if it appears in a document 3 times). Literal matching (e.g., taking inner products of document vectors with the query vector and then choosing the maximum) is not usually the best strategy because latent relationships and document clusters are not revealed.

Instead, the latent semantic indexing (LSI) method (Deerwester et al, 1990; Berry et al, 1995), approximates the document matrix (consisting of all document vectors) via a low-rank approximation and uses the columns of the low-rank matrix to compute the inner products with the query vector.

We form a low-rank approximation to the training chromosomes matrix X using the singular value decomposition (SVD)

$$X = U\Sigma V^T = \sum_{i=1}^{L} \sigma_i u_i v_i^T,$$

where $U = [u_1\, u_2 \ldots u_L]$ is an $L \times L$ orthogonal matrix, $V = [v_1\, v_2 \ldots v_M]$ is an $M \times M$ orthogonal matrix, and $\Sigma = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_L\}$ is a diagonal matrix of size $L \times M$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_L \geq 0$ where we assume $L \leq M$. The best rank-R ($R < L$) approximation of A is given by the truncated SVD

$$\bar{U}\bar{\Sigma}\bar{V}^T = \sum_{i=1}^{R} \sigma_i u_i v_i^T \qquad (1)$$

(Golub and Van Loan, 1996). Since it filters out much of the variations, a low-rank approximation often works better than the original matrix in an application such as this. If all chromosomes of a given type matched identically, then X would have rank 24 (i.e., there would be only 24 unique columns), so a reasonable value for R is 24.

We apply LSI to the chromosome problem as follows. Suppose y is the data for a chromosome of unknown type that has already been stretched to length L. (If y is longer than L, then the number of entries is reduced using rectangular rule integration.) Then, using the truncated SVD approximation of X, an M-vector of scores is computed by

$$s = \sum_{i=1}^{R} \sigma_i (y^T u_i) \bar{D} v_i,$$

where $\bar{D}$ is a diagonal matrix that normalizes the columns of $\bar{U}\bar{\Sigma}\bar{V}^T$ to Euclidean length 1. Ideally, the unknown chromosome has the identity corresponding to the largest score s_j, $j = 1, 2, \ldots, M$. However, our training data is not perfect, and using only the top

scoring chromosome may cause anomalous results. Instead, we use the top T scores and a "voting scheme" as follows. For $t = 1, 2, \ldots, T$, if the tth biggest score is from a chromosome of type k, add

$$\frac{T - t + 1}{M_k}$$

to type k's tally, where M_k is the number of training chromosomes of type k. A typical value of T is 5. The unknown chromosome's type is identified as the type with the highest tally.

It is relatively easy to update the SVD if new chromosomes are added to the training set (Gu and Eisenstat, 1993), and this is a major advantage.

Principal Component Analysis

The SVD algorithm implicitly assumes that the measurements are independent and have similar standard deviations. If we wish to take covariances into account, then we need to use the SVD in a somewhat different way. Rather than finding the chromosome type with the largest score, we find the type with the minimal Mahalanobis distance (defined in Equation 2) to the mean of the training vectors of that type. Let X_k be the matrix whose columns are training chromosomes of type k, normalized to Euclidean length one, and let $\bar{x}_k$ be the mean of these column vectors. Define the matrix

$$\tilde{X}_k = X_k - \bar{x}_k e^T,$$

where e is the vector of all 1's. Let $U_k \Sigma_k V_k^T$ be $\tilde{X}_k$'s SVD. Then the Mahalanobis distance between an unknown chromosome y (stretched to length L and normalized to Euclidean length one) and the cluster for chromosome type k is

$$d_k = (y - \bar{x}_k)^T (\tilde{X}_k \tilde{X}_k^T)^{-1} (y - \bar{x}_k) = \|\Sigma_k U_k^T (y - \bar{x}_k)\|^2. \quad (2)$$

This is often simplified by assuming a common covariance matrix $\tilde{X}\tilde{X}^T$, a possibly low-rank matrix derived from the full set of training chromosomes, in place of $\tilde{X}_k \tilde{X}_k^T$. The unknown chromosome's type is identified as the type with the least distance to its cluster. See Jackson (1991).

Fisher Discriminant Analysis

In this algorithm (Mardia et al, 1979), the training phase first computes two matrices:

$$S_w = \sum_{k=1}^{24} \tilde{X}_k \tilde{X}_k^T,$$

where $\tilde{X}_k$ was defined for PCA and

$$S_b = \sum_{k=1}^{24} M_k (\bar{x}_k - \mu)(\bar{x}_k - \mu)^T,$$

where $\bar{x}_k$ is also as defined for PCA and M_k is again the number of training vectors of type k. Here the vector μ is the "grand mean"; that is, the mean of all M training chromosomes. We then find a 23-dimensional subspace over which the minimum of the function

$$\frac{x^T S_b x}{x^T S_w x}.$$

is maximized. This will determine the projection subspace which maximizes the ratio of the *between cluster* distances to the *within cluster* distances. The solution can be computed by solving a generalized eigenvalue problem. If the columns of $V = [v_1, v_2, \ldots, v_{23}]$ form an basis for this subspace, then for any vector x, the Fisher discriminant vector (or projection) is given by $V^T x$. We use this matrix to project each of the 24 training clusters and then compute a common covariance matrix based on the projected training data.

An unknown chromosome y is then classified by first projecting its vector into the 23 dimensional space (i.e., $V^T y$) and then computing the distance to the nearest cluster using a procedure analogous to the PCA method. Note that unlike PCA, we use the projected covariance matrix rather than a low rank approximation of it.

Hidden Markov Models

The HMMs that we build produce a sequence of *observations* from various *states.* The output of the model observed at time t is denoted by O_t. An observation consists of either a single gray level value, $O_t = \{z_t\}$; a gray level value and its first difference, $O_t = \{z_t, z_t'\}$; or a gray level value, its first difference, and its second difference, $O_t = \{z_t, z_t', z_t''\}$. For convenience, the rest of this discussion will assume a triple of observations.

The HMM consists of a Markov chain that generates output typical of the vectors of a particular type of chromosome. The state of the chain is hidden from the observer, hence the name of the method.

To prepare the data to create an HMM, we do not need to make all of the vectors equal in length (as in SVD and PCA), but we do normalize each sequence to mean zero and standard deviation one by subtracting the mean of each of the three components and dividing by the standard deviation of this component. This normalization is done on a sequence-by-sequence basis, i.e., individual means and standard deviations are computed for each sequence. Thus, it removes some of the variability caused by inconsistencies in staining and illumination.

For chromosome karyotyping we create 24 HMMs, one per chromosome. We interpret the *states* of the HMM as positions in the sequence of gray level triples for an idealized chromosome of this type and the *observation* from a given state as a typical sample of gray level triples at this position. We first specify the ideal number of states (i.e., length) and denote it by N. The HMM is denoted as $\lambda = (A, B, p)$, and the meaning of the parameters is described below.

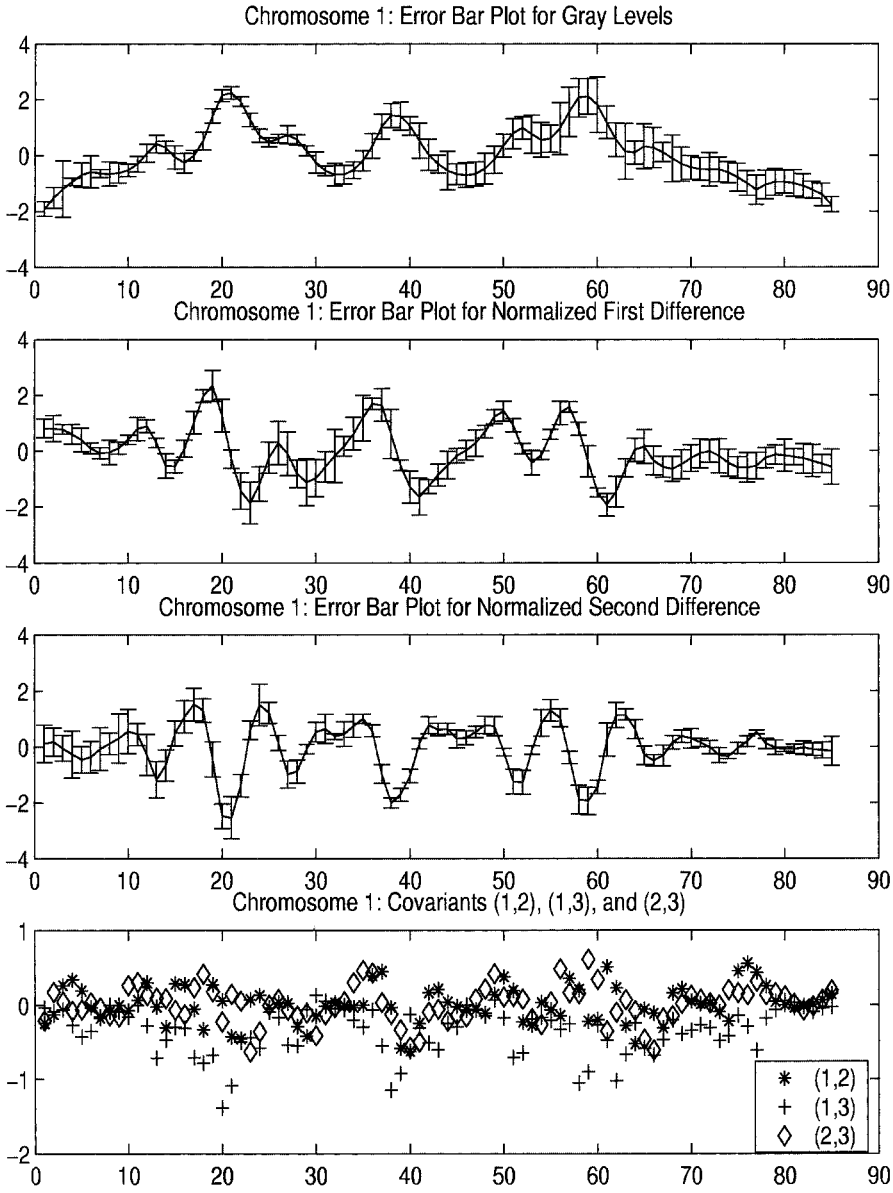

Figure 4.
HMM Output (*B*) Matrix for chromosome 1. Please see the text for detailed descriptions of these graphs.

- We set the number of states N in the model to the median length of the vectors for all the chromosomes of this type in the training data.
- From a given state, we allow transitions only to the same state, the next one, or the one beyond that. This allows for "insertions" and "deletions" but no backtracking. We set these probabilities to be the same for all but the last states so that the transition matrix A is defined by exactly 2 parameters: a_{next}, the probability of moving to the next state, and a_{skip}, the probability of moving to the one beyond that. (We cannot "skip" out of state $N - 1$, so the probability of moving to state N from there is set to $a_{\text{next}}/(1-a_{\text{skip}})$. The probability of staying in state $N - 1$ is similarly adjusted.) The probability of staying in a state a_{stay} is defined to be 1 minus the probability of moving out of it.
- The output function B for state i ($i = 1, \ldots, N$) is specified by a mean (3 values) μ_i and covariance (a symmetric 3×3 matrix) C_i. The probability of outputting a gray scale triple x in state i is modeled as a mixture of two 3-dimensional normal distributions: one mean zero variance I (the identity matrix) and one estimated from the training data given by μ_i and C_i. The mixture model is used to lessen the effects of overtraining, and the mixing weight r was fixed at 0.8. Thus, the observation triple x is modeled as $x \sim r\mathcal{N}(\mu_i, C_i) + (1 - r)\mathcal{N}(0, I)$ (where $\mathcal{N}$ denotes the Normal distribution with given mean and covariance), and the probability b_i of observing ζ is proportional to

$$\frac{r}{\sqrt{|C_i|}} e^{-(\zeta-\mu_i)^T C_i^{-1}(\zeta-\mu_i)} + (1-p)e^{-\zeta^T\zeta}, \qquad (3)$$

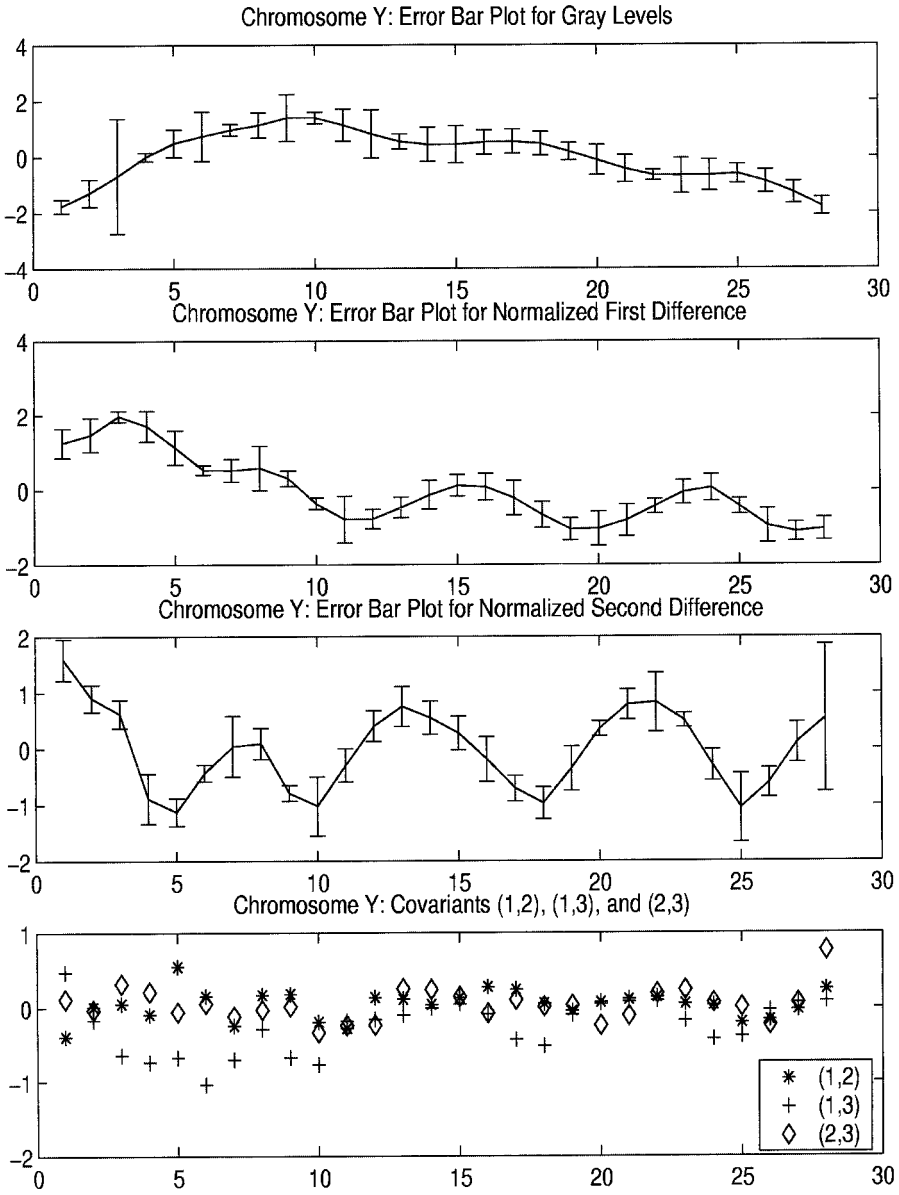

Figure 5.
HMM Output (*B*) Matrix for chromosome Y. Please see the text for detailed descriptions of these graphs.

where $|C_i|$ is the determinant of the matrix.

- The *i*th entry in the vector p ($i = 1, \ldots, N$) gives the probability of beginning with the model in state *i*. Initially, all probabilities are equal; i.e., $p(i) = 1/N$ for $i = 1, \ldots, N$.

The hidden Markov model is illustrated in Figure 3. The $10N + 3$ parameters that define $\lambda = (A, B, p)$ for the model are chosen by using Baum-Welch re-estimation (Baum et al, 1970) to train over the data for that chromosome. This optimization maximizes the likelihood that the model is correct.

Once an HMM for a particular chromosome type is constructed, a sample chromosome can be generated as follows. Choose a starting state. The next state is chosen according to probabilities as specified in a transition probability matrix, *A*. Continue until we get to state *N*. This may take more or fewer than *N* transitions depending on how many *skip*'s and *stay*'s we have. The sequence of observations generated in this process is typical of this type of chromosome.

In Figure 4 we show examples of the output parameters for the model for chromosome 1 for the Copenhagen data set. The horizontal axis is the index of the state in the HMM (i.e., the position in the sequence of gray levels). We plot the mean gray scale output from that state, displaying the variance by an error bar. There are three plots: one for the gray level value, one for the first difference, and one for the second difference. The covariances between the gray level and its first difference (components 1 and 2), between the gray level and its second difference (components 1 and 3), and finally between the first difference and the second difference (components 2 and 3) are given in

the fourth plot. Figure 5 gives the same information for chromosome Y.

Baum-Welch Re-estimation for HMM

To build an HMM, we use Baum-Welch Re-estimation to determine the transition matrix A, the output function B, and the initial state distribution p. We also use it for scoring chromosomes.

For an individual chromosome, the Baum-Welch method aligns the gray level triples to an idealized model sequence so that common features can be statistically exposed. Specifically, we start with the nominal HMM $\lambda = (A, B, p)$ and a sequence of observations $\{O_1, O_2, \ldots, O_T\}$ for the m-th chromosome of type k. We compute the N-vectors $\alpha_t^{(m)}$, $\beta_t^{(m)}$, $\gamma_t^{(m)}$ and $N \times N$ matrices $\xi_t^{(m)}$ for $t = 1 \ldots, T$, which are probability distributions that give the alignment to the idealized chromosome we will shortly define. For convenience, we drop the superscript until we show how to combine the individual parameters to form an HMM for type k.

We let $\alpha_t(i) = \Pr(\{O_1, O_2, \ldots, O_t\}$ and state $i|\lambda)$ for $t = 1, \ldots, T$. In other words, $\alpha_t(i)$ is the probability that we have observed the sequence $\{O_1, O_2, \ldots, O_t\}$ and are currently in state i ($1 \le i \le N$) given the model. We can compute $\alpha_t(i)$ recursively as follows. Let $\alpha_1(i) = p(i)$ and compute

$$\alpha_t = D_{O_t} A^T \alpha_{t-1} \text{ for } t = 2, \ldots, T,$$

where

$$D_{O_t} = \text{diag}\{b_1(O_t), b_2(O_t), \ldots, b_N(O_t)\},$$

where $b(O_t)$ is as defined in Equation 3 with $\zeta = O_t$. The probability of the entire observation sequence is given by

$$\Pr(\{O_1, O_2, \ldots, O_T\}|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \qquad (4)$$

We define $\beta_t(i) = \Pr(\{O_{t+1}, O_{t+2}, \ldots, O_T\}|i$ and $\lambda)$, i.e., $\beta_t(i)$ is the probability that we will observe the sequence $\{O_{t+1}, O_{t+2}, \ldots, O_T\}$ given that we are at state i and λ. A backwards recursion lets us compute $\beta_t(i)$ as follows. Initialize β_T to all ones, and then

$$\beta_t = A D_{O_{t+1}} \beta_{t+1} \text{ for } t = T - 1, \ldots, 1.$$

The results of these two recursions are combined to form

$$\gamma_t(i) = \Pr(i \text{ at } t|\{O_1, O_2, \ldots, O_T\} \text{ and } \lambda),$$

i.e., $\gamma_t(i)$ is the probability of being in state i at time t given the sequence of observations $\{O_1, O_2, \ldots, O_T\}$ and the model λ. The formula is given by

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\Pr(\{O_1, O_2, \ldots, O_T\}|\lambda)}.$$

Likewise, $\xi_t(i, j) = \Pr((i \text{ at } t)$ and j at $t + 1)|\{O_1, O_2, \ldots, O_T\}$ and $\lambda)$ is the probability of being at state i at time t and at state j at time $t + 1$ given the sequence of observations $\{O_1, O_2, \ldots, O_T\}$ and λ, and is given by

$$\xi_t(i, j) = \frac{\alpha_t(i) A_{ij} \beta_{t+1}(j) b_j(O_{t+1})}{\Pr(\{O_1, O_2, \ldots, O_T\}|\lambda)}.$$

The probabilities given by the γ's and the ξ's are used to derive new estimates for the parameters of the HMM. Given the ξ's, a new estimate of the Markov transition matrix is computed as follows. Compute

$$\tilde{a}_{\text{stay}} = \sum_{m=1}^{M_k} \sum_{i=1}^{N} \sum_{t=1}^{T} \xi_t^{(m)}(i, i),$$

$$\tilde{a}_{\text{next}} = \sum_{m=1}^{M_k} \sum_{i=1}^{N-1} \sum_{t=1}^{T} \xi_t^{(m)}(i, i+1),$$

$$\tilde{a}_{\text{skip}} = \sum_{m=1}^{M_k} \sum_{i=1}^{N-2} \sum_{t=1}^{T} \xi_t^{(m)}(i, i+2).$$

The above three quantities are normalized to sum to 1 and thereby give the new Baum-Welch estimate of A for the k-th chromosome type.

Similarly, we re-estimate the transition output function B. Recall that B is a mixture of two 3 dimensional normal distributions: one is fixed to be mean zero and covariance I, and the other is parameterized by μ_i and C_i. In order to account for the contributions of each sequence in the training set, we compute the parameters for a new estimate for the output function B as

$$\hat{\mu}_i = \frac{\sum_{m=1}^{M_k} \sum_{t=1}^{T} \gamma_t^{(m)}(i) O_t^{(m)}}{\sum_{m=1}^{M_k} \sum_{t=1}^{T} \sum_{i=1}^{N} \gamma_t^{(m)}(i)} \text{ for } i = 1, \ldots, N, \qquad (5)$$

and

$$\hat{C}_i = \frac{\sum_{m=1}^{M_k} \sum_{t=1}^{T} \gamma_t^{(m)}(i)(O_t^{(m)} - \mu_i)(O_t^{(m)} - \mu_i)^T}{\sum_{m=1}^{M_k} \sum_{t=1}^{T} \sum_{i=1}^{N} \gamma_t^{(m)}(i)} \text{ for } i = 1, \ldots, N. \qquad (6)$$

Finally, the initial state distribution is re-estimated for each chromosome of type k by setting

$$p^{(m)} = \gamma_1^{(m)}.$$

This completes one iteration of Baum-Welch re-estimation. The iterations continue until the relative improvement in log probability of the training data is less than 10^{-3}.

Scoring with HMM

Once we have defined these 24 HMMs, we can classify an unknown chromosome by first calculating the probability of obtaining its observations as the output from each model. To compute a score from model k ($k = 1, \ldots, 24$),

- We perform two iterations of Baum-Welch re-estimation (Baum et al, 1970), updating only the initial state distribution p.
- We then compute the probability that the observations were an output of each model using Equation 4.

The 24 resulting scores are correlated, for example, with chromosomes of nearly the same length rather likely to be confused. Correlations in the scores can be exploited by viewing the 24 scores as feature vectors. The classification is then achieved by building a linear discriminant analysis model based on the training data using these 24 long vectors. The LDA score based on the 24 HMM scores is then used to classify the chromosome.

References

Baum LE, Petrie T, Soules G, and Weiss N (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41:164–171.

Berry MW, Dumais ST, and O'Brien GW (1995). Using linear algebra for intelligent information retrieval. SIAM Review 37:573–595.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, and Harshman R (1990). Indexing by latent semantic analysis. J Soc Inform Sci 41:391–407.

Golub GH and Van Loan CF (1996). Matrix computations, 3rd ed. Baltimore: Johns Hopkins University Press.

Gu M and Eisenstat S (1993). A stable and fast algorithm for updating the singular value decomposition. New Haven: Yale University Department of Computer Science. RR-939.

Jackson JE (1991). A user's guide to principal components. New York: John Wiley and Sons.

Mardia KV, Kent JT, and Bibby JM (1979). Multivariate analysis. New York: Academic Press.