Consiglio Nazionale delle Ricerche
ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE
PISA
Pubblicazione A74-17

M. Aiello, C. Lami, U. Montanari.

# Optimal Matching of Wheat Chromosomes

Estratto da: Computer Graphics and Image Processing, 3 (1974), 225-235.

COMPUTER GRAPHICS AND IMAGE PROCESSING (1974) 3, (225–235)

# Optimal Matching of Wheat Chromosomes

MARIO AIELLO, CARLO LAMI, AND UGO MONTANARI

*Istituto di Elaborazione dell'Informazione del CNR, Via S. Maria 46, 56100 Pisa, Italy.*

*Communicated by A. Rosenfeld*

*Received December 1, 1973*

A program for automatic analysis of wheat chromosomes is presented whose last stage recognizes homologous pairs by means of an optimal matching procedure in the pattern space.

## 1. INTRODUCTION

A number of programs for chromosome analysis have been developed recently by many groups in different countries as examples of computer applications to biomedical research (see for instance [1,2]). The program described here is different from most of those in the following aspects:

(a) Here we analyze spreads of *wheat* chromosomes, which are observed at an early stage of the mitotic phase, so that chromosomes appear as sticks (see Fig. 1). The final goal of the project is to examine the effects of radiations for isolating useful mutations[1] [3–5].
(b) The program is organized to allow the user to specify the set of measures and the metric of the pattern space he finds the most appropriate for his problem.
(c) The last part of the program consists of an optimal matching procedure, based on a heuristic search technique, which recognizes homologous pairs.

Chromosome spreads are photographed and then digitized with a flying spot scanner designed and built at our Institute [6,7]. The final output of the program is a printout of the karyotype.

In this paper, we give special attention to the optimal matching phase. In particular, it is shown that the heuristically guided matching algorithm is sensitive to the order of data presentation. A simple algorithm for rearranging data in a suitable way is then described.

## 2. CHROMOSOME SEGMENTATION AND ANALYSIS

The process of analyzing a metaphase is logically divided into three steps: segmentation, parameter evaluation, and optimal matching. In this section

[1] This project is being performed in collaboration with the Institute of Agricultural Science of the University of Pisa.

FIG. 1. A wheat chromosome spread.

we outline the first two steps, while the last step will be described in Sections 3 and 4.

In the segmentation phase, objects are first isolated by means of an edge follower, assuming that the background consists of points with gray values lower than a given threshold. At this stage some filters are established which reject spurious objects on the basis of parameters roughly characterizing the shape and density distribution. Each chromosome is then reduced by a factor of three and processed separately.

Touching chromosomes are then recognized and separated by means of a very simple-minded algorithm which uses an increasing threshold for dividing two chromosomes. This method assumes that connectivity breaks at the touching point first, and at the centromeric constriction later.

Bent chromosomes are then discriminated on the basis of partial profiles referred to the principal axis of inertia with minimal second-order moment[2]. A "straightening" procedure is then applied, which consists of finding an axial line to which we refer all the measures. Such a procedure consists essentially of four steps (for a detailed description, see [4]):

*Step 1.* Divide the chromosome in two regions $A$ and $B$ using the normal line to the inertia axis $ss'$ at a point $P$ roughly corresponding to the minimum of the internal profile and to the maximum of the external profile (see Fig. 2).

*Step 2.* Compute the principal axes of inertia $aa'$ and $bb'$ of regions $A$ and $B$ respectively. Find the bisectrix $cc'$ of the angle between $aa'$ and $bb'$ which has the centers of gravity of regions $A$ and $B$ on opposite sides. From

[2] The density profiles are obtained by projecting chromosome points on a symmetry axis and summing up gray values. Partial profiles use only points on one side of the axis.

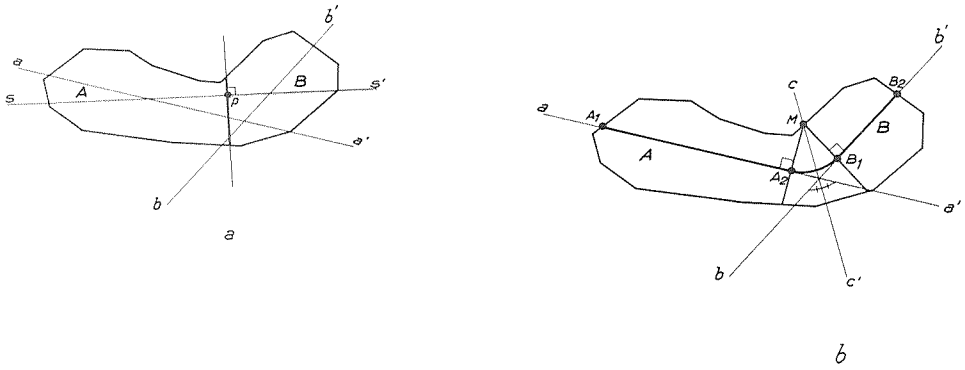


FIG. 2. An algorithm for determining an axial line in a bent chromosome.

the internal intersection $M$ between $cc'$ and the contour, trace the normal lines to axes $aa'$ and $bb'$, thus obtaining two new "straight" regions $A$ and $B$.

*Step 3.* Iterate step 2 until the partial profiles of each region match satisfactorily in their peaks and valleys. However, after a fixed number of iterations stop with an error message.

*Step 4.* Trace an arc of a circle with center in $M$ to join axes $aa'$ and $bb'$. The segment $A_1A_2B_1B_2$ is the computed axial line.

In the parameter evaluation phase, the total and partial profiles referred to the axial line are evaluated. By examining the total profile, it is possible to determine the centromere position (the absolute minimum) and secondary constrictions (local minima), if present. At this stage of the computation, the user is allowed to accomplish whatever measure he finds proper. He can define new procedures which have access to all the results so far obtained.

As mentioned above, all these algorithms are applied to reduced copies of chromosomes in order to minimize the processing time. When a failure is detected (essentially in the algorithm for dividing two touching chromosomes or in the algorithm for "straightening" the bent ones), the same algorithm is applied to a full-size chromosome. In this way it is possible to save much computation time with equally good results.

## 3. THE MATCHING PROBLEM

The parameters computed in the previous phase are used for recognizing homologous pairs. We call this operation *chromosome matching*. Matched chromosomes are then presented in the form of a printed karyogram.

As is usual in pattern recognition, after the measurement phase chromosomes are represented as points of a multidimensional space. If a suitable metric is assumed in this space, we can define the *cost* of a particular matching as the sum of the distances between the points in the pair. The suggested matching is the matching that minimizes such a cost.

Following the approach described in the introduction, our program provides only a general matching algorithm, by leaving to the user the possibility of specifying details by experimenting. In fact, the user must provide to

the matching algorithm a square matrix containing the distances between all possible chromosome pairs. Such distances are computed after a suitable measure normalization or possibly after a space distortion taking into account the statistics of similar solved cases.

The distance matrix mentioned above can be considered to express the weights on the arcs of a complete graph $H$: therefore we are driven to consider the so-called *graph weighted matching problem.* This problem was approached by Edmonds, who gave an algorithm running in polynomial time, roughly $n^4$ in the worst case [8]. Edmonds' algorithm is quite efficient in general, but (as argued in [9]) an even faster, possibly approximated technique is needed in our case. Therefore we have developed a method taking advantage of the fact that chromosome data are not random, but tend to have at least some pairs immediately recognizable. For instance, we capitalize on the fact that in many cases the nearest chromosome to a given one is its actual partner in the optimal matching.

Our approach to the graph matching problem is to translate it into a shortest path problem in a graph $G$ and then to direct the search using a heuristic technique. According to the time and storage available, it is possible to seek the optimal solution or simply a good one. Nodes of graph $G$ have a partial solution of the matching problem associated with them, where some even subset of the vertices of $H$ is matched (it does not matter how) and the rest of them are not. Thus if $H$ has $n$ nodes, the vertices of $G$ can be partitioned into $(n/2) + 1$ levels, according to the cardinality of the associated set. The arcs of $G$ connect only pairs of nodes in adjacent levels. Each of them corresponds to (and has the cost of) the matching of a further pair. As an example, we can see in Fig. 3 the graphs $H$ and $G$ for $n = 6$. Note that every subset of matched vertices is represented by a bit string that encodes the characteristic function of such a subset. In particular, nodes $V_a$ and $V_z$ correspond to the initial state (no chromosome is matched) and to the final state (all chromosomes are matched), respectively.

It should be clear that every path in graph $G$ corresponds to a particular matching in $H$. For instance, in Fig. 3 the path (000000), (100001), (110101), (111111) has the cost $w_{16} + w_{24} + w_{35}$ and corresponds to the matching (1,6), (2,4), (3,5). However, it is interesting to notice that if all the possible subsets of a given cardinality were present at a level, then the correspondence between paths in $G$ and matchings in $H$ would not be one-to-one since many paths would correspond to the same matching. In fact, given a matching, permuting the order of the pairs would result in the same matching but in different paths. To allow a one-to-one correspondence (see [9] for a proof), at the $k$th level exactly those sets of cardinality $2k$ are present, which contain the first $k$ elements. Furthermore, every vertex $V_i$ of $G$ in the $k$th level is adjacent to every vertex $V_j$ of the $(k+1)$th level such that the subset represented by $V_j$ can be obtained from the subset represented by $V_i$ by adding the *first* chromosome not present in $V_i$ and any other chromosome. For instance, in the example in Fig. 3, we can see that vertex (101101) is not present in the second level, while vertices (110000) of the first level and 110110 of the second level are not connected by an arc.

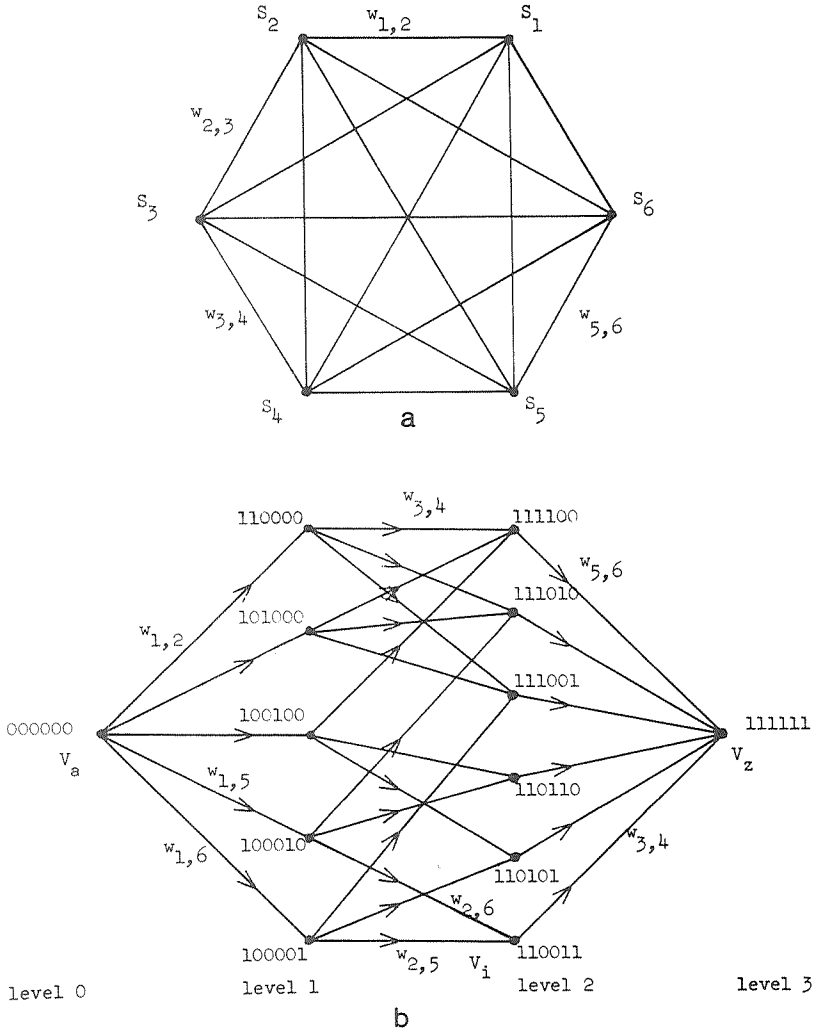


FIG. 3. A complete weighted graph $H$ with six nodes (a) and its corresponding weighted graph $G$ (b).

With the above assumptions, it is possible to prove [9] that the number of vertices in $G$ is $F_{n+1}$, where

$$F_i = \phi^i \sqrt{5}$$

rounded to the nearest integer, $i = 1,2, \ldots$, with

$$\phi = (1 + \sqrt{5})/2 = 1.61803$$

Integers $F_i$ are known as the Fibonacci numbers.

## 4. SEARCHING FOR AN OPTIMAL MATCHING

Searching for a shortest path from $V_a$ to $V_z$ in a graph $G$ is a well-studied problem in operations research and artificial intelligence. Here we use an

algorithm due to Hart, Nilsson, and Raphael [10,9] which is based on a heuristic estimate and which can be seen as a merging of branch and bound and dynamic programming. In this algorithm, at any intermediate stage, the vertices of $G$ are partitioned into three sets: *closed* vertices, *open* vertices, and *blank* vertices. To a closed or open vertex $V_i$, a parameter $p_i$ is associated, which expresses the length of some path from $V_a$ to $V_i$. Furthermore, if $V_i$ is closed, parameter $p_i$ expresses the length of a *shortest* path in $G$ from $V_a$ to $V_i$, i.e., the distance $d_{ai}$.

A second parameter $q_i$ is also available for *every* vertex $V_i$. This parameter must be provided from sources external to the search and represents a lower estimate of the shortest path length from $V_i$ to $V_z$; namely:

$$q_i \leqslant d_{iz},\ i = 1, \ldots, n.$$

Furthermore, a triangular property must hold:

$$q_i \leqslant d_{ij} + q_j,\ i,j = 1, \ldots, n.$$

The meaning of parameter $q_i$ is easily explained in the special case in which graph $G$ is a road map: It is clear that a lower bound to the path length can be easily computed as the straight-line distance, and it can be readily understood how such a bound allows one to exclude many hopeless guesses.

From the two parameters $p_i$ and $q_i$ we can compute, for every *open* vertex, the *total estimate*

$$f_i = p_i + q_i$$

which is an estimate of the total distance $d_{az}$. The iterative step of our algorithm consists of:

(i) Determining an open vertex $V_k$ whose total estimate is minimal and closing it;

(ii) updating the value of all *open* vertices $V_j$ adjacent to $V_k$ using the formula

$$\bar{p}_j = \min\ (p_j, p_k + t_{kj}),$$

where $t_{kj}$ is the weight of the arc $V_kV_j$;

(iii) opening all *blank* vertices $V_j$ adjacent to $V_k$ by assigning them the parameter

$$p_j = p_k + t_{kj}.$$

This step is performed until the final vertex becomes closed. It is then possible to prove [10] that the value $p_z$ will indeed give the shortest-path-length $d_{az}$.

It is clear that search directionality is highly dependent upon estimate accuracy. In particular, it is possible to see that if the estimate is exact, all the vertices closed by the algorithm[3] will be on some optimal path.

[3] Note that the total number of closed vertices coincides with the number of steps of the algorithm.

In our chromosome problem, a suitable estimate (see the proof in [9]) is given by the following formula:

$$q_i = \frac{1}{2} \sum_{S_h \notin T_i} \min_{\substack{S_k \notin T_i \\ h \neq k}} w_{hk}. \tag{4.1}$$

Here $T_i$ is the set of matched chromosomes corresponding to vertex $V_i$ of $G$, while $w_{hk}$ is the entry of the distance matrix of graph $H$ corresponding to chromosomes $S_h$ and $S_k$.

In Fig. 4a we see a two-dimensional pattern space for the chromosome spread in Fig. 1, where mass and centromeric index are used as measures. The sum of the lengths of the segments shown in Fig. 4b gives the estimate (4.1) for the initial node $V_a$. In Fig. 4c we see the optimal matching instead, whose cost is given again by the segment total length. By comparing Figs. 4b and 4c we see in fact how the cost of many "easy" pairs is exactly guessed by the estimate.

In Fig. 5 we see the final karyogram: matched chromosomes are rotated into a vertical position and printed (with superimposed characters to reproduce gray levels) a pair per page.

## 5. EXPERIMENTAL RESULTS

The computer program described in this paper has been implemented in FORTRAN IV on the IBM 360/67 computer of the National University Com-

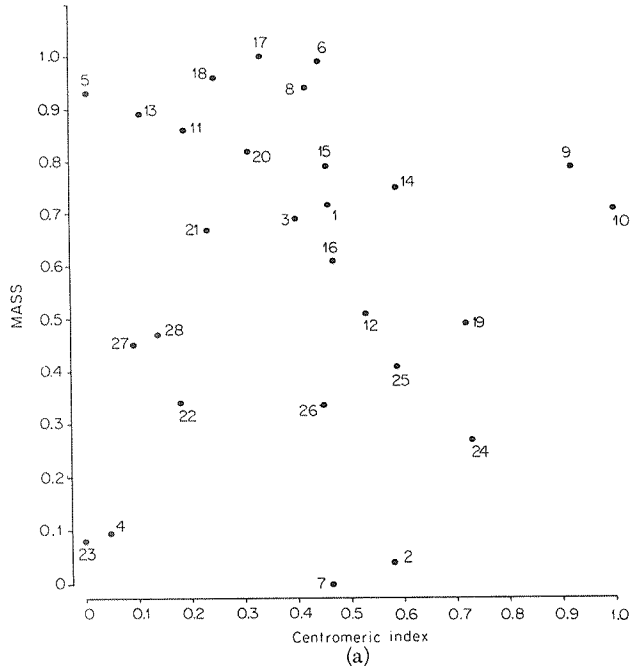


FIG. 4. Matching the chromosomes in Fig. 1. (a) The pattern space; (b) the heuristic estimate $q_a$ for the initial vertex $V_a$ of graph $G$; (c) the optimal matching.

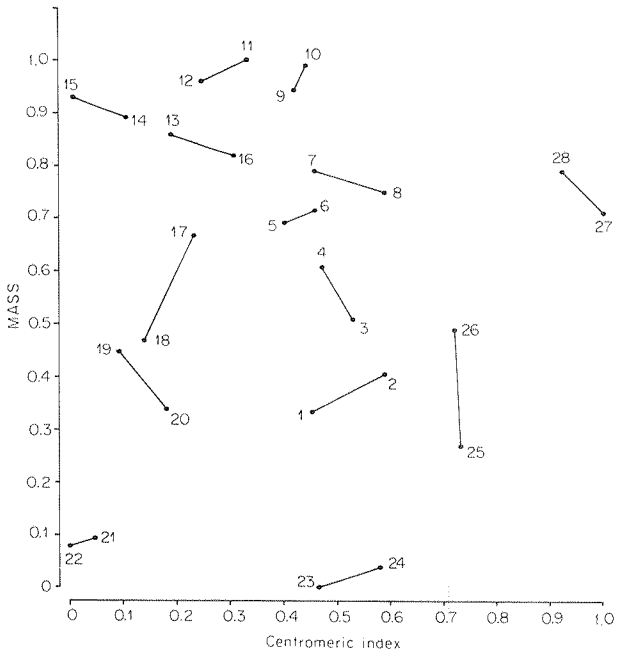


FIG. 6. Ordering data for the matching algorithm.

eral, loosely speaking, our algorithm is faster if all "reasonable" partners of each chromosome are not too far from it in the given ordering.

According to our experience, we have found it convenient to introduce a reordering step before optimal matching, which operates according to the following algorithm.

*Algorithm REORD*

*Step 1.* Let $[a_{ij}]$ be a matrix whose $(i,j)$ entry is the distance between chromosome $i$ and chromosome $j$ in the pattern space;
select in some way a chromosome $i$ and mark it;
let $r = 1$, $p_r = i$, and $t = 0$.

*Step 2.* If all chromosomes are marked, go to step 3;
find the unmarked chromosome $k$ which is the nearest to $i$, namely such that:

$$a_{ik} = \min_{j \text{ is unmarked}} a_{ij};$$

mark chromosome $k$ and let $r = r + 1$, $p_r = k$, $t = t + a_{ik}$, and $i = k$;

go to step 2.

*Step 3.* Print $t$ and $p_s$ $(s = 1, \ldots, r)$;
stop.

[5] An algorithm is called *greedy* if at every step it makes the best local choice and has no possibility of undoing a choice which later proves to be globally unsatisfactory.

It is easy to see that algorithm REORD finds in a sense the shortest hamiltonian chain $p_1p_2, \ldots, p_r$ in graph $H$ which can be computed with a "greedy" algorithm[5]. A slightly better result can be obtained by repeating algorithm REORD for all possible selections of chromosome $i$ in step 1 and by choosing a chain with minimal total length $t$. For example, the ordering in Fig. 6 has been obtained in this way, with a total computing time of 0.8 seconds.

ACKNOWLEDGMENTS

The authors thank Professor A. Grasselli for conceiving the system; G. Levi, N. Lijtmaer, and A. Albano for working at earlier stages of the project; L. Azzarelli and R. Panicucci for developing the SADAF flying spot system; and Professor F. D'Amato of the Institute of Genetics of the School of Agricultural Science of the University of Pisa for providing the testing material.

REFERENCES

1. R. S. LEDLEY, Some clinical applications of pattern recognition, *Proc. 1st IJCPR, Washington, D. C.;* October 30–November 1, 1973, pp. 89–112.
2. D. RUTOVITZ, Automatic chromosome analysis, *Brit. Med. Bull.* **24**, 1968, 260–267.
3. G. LEVI, AND N. LIJTMAER, Un metodo per la classificazione automatica dei cromosomi, *Calcolo, Vol. V*, Supp. No. 1, pp. 1–31 (1968).
4. M. AIELLO, A. ALBANO, AND G. LEVI, A procedure for determining the centromere in the classification of wheat metaphases by digital computer, *Computers and Biomedical Research* **4**, 1970, 330–343.
5. M. AIELLO, C. LAMI, AND U. MONTANARI, A system for computer measurements and karyotyping of wheat metaphases, *Proc. 1st IJCPR, Washington, D. C.*, October 30–November 1, 1973, pp. 205–219.
6. L. AZZARELLI, AND R. PANICUCCI, Descrizione del sistema S.A.D.A.F. per la lettura e digitalizzazione di fotogrammi, I.E.I. Nota Tecnica C72–2, May 1972.
7. C. CARLESI, AND U. MONTANARI, Manuale d'uso dei programmi di utilità del sistema S.A.D.A.F. per la lettura e digitalizzazione di fotogrammi, I.E.I. Nota Tecnica C73-11, December 1973.
8. J. EDMONDS, Maximum matching and a polyhedron with 0,1 vertices, *J. Res. NBS (Math. and Math. Phys.)* **69B**(1,2), Jan.–June 1965, pp. 125–130.
9. U. MONTANARI, Heuristically guided search and chromosome matching, *Artificial Intelligence* **1**, 1970, 227–245.
10. P. HART, N. J. NILSSON, AND B. RAPHAEL, A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. SCC* **4** (2), 1968, 100–107.
11. M. AIELLO, C. LAMI, AND U. MONTANARI, Documentation of package CHROMO for the automatic analysis and karyotyping of wheat chromosomes, I.E.I. Nota Tecnica C72–8, December 1972.