ODS2: A Multiplatform Software Application for Creating Integrated Physical and Genetic Maps

David Hall,*,† Suchendra M. Bhandarkar* and Jian Wang†

**Department of Computer Science, †Department of Genetics, The University of Georgia, Athens, Georgia 30602-7404*

Manuscript received September 1, 2000
Accepted for publication December 8, 2000

ABSTRACT

A contig map is a physical map that shows the native order of a library of overlapping genomic clones. One common method for creating such maps involves using hybridization to detect clone overlaps. False-positive and false-negative hybridization errors, the presence of chimeric clones, and gaps in library coverage lead to ambiguity and error in the clone order. Genomes with good genetic maps, such as *Neurospora crassa*, provide a means for reducing ambiguities and errors when constructing contig maps if clones can be anchored with genetic markers to the genetic map. A software application called ODS2 for creating contig maps based on clone-clone hybridization data is presented. This application is also designed to exploit partial ordering information provided by anchorage of clones to a genetic map. This information, along with clone-clone hybridization data, is used by a clone ordering algorithm and is represented graphically, allowing users to interactively align physical and genetic maps. ODS2 has a graphical user interface and is implemented entirely in Java, so it runs on multiple platforms. Other features include the flexibility of storing data in a local file or relational database and the ability to create full or minimum tiling contig maps.

A contig map is a physical map that shows the native order of a library of overlapping genomic clones. Such maps help in the positional cloning of genes, serve as a framework for whole genome sequencing (Chumakov *et al.* 1995; McPherson 1997), and are used in the study of the large-scale organization of genomes (Prade *et al.* 1997). Approaches that have been used to infer the order of libraries of clones include fingerprinting, assaying for sequence-tagged sites (STSs), and direct detection of clone overlaps. Clones may be fingerprinted by treatment with restriction enzymes and measurement of the sizes of the resulting fragments (Coulson *et al.* 1986; Olson *et al.* 1986). Another fingerprinting method is based on hybridization of oligonucleotide probes (Lehrach *et al.* 1990). Sequence-tagged sites can be detected with the polymerase chain reaction (Green and Olson 1990). Hybridization (Prade *et al.* 1997) and DNA sequencing (Venter *et al.* 1996) can be used to detect clone overlaps directly.

There are a number of types of errors associated with these protocols that make recovery of the native clone order difficult. False-positive and false-negative PCR and hybridization results are common. For example, false-positive and false-negative error rates for the *Aspergillus nidulans* mapping project (Prade *et al.* 1997), which used hybridization to detect clone overlaps, were estimated to be 0.5 and 10%, respectively (R. A. Prade, J. Griffith, K. Kochut, J. Arnold and W. E. Timberlake, personal communication). In restriction enzyme fingerprinting, gel electrophoresis is commonly used to measure the size of restriction fragments. A problem with this method is that different fragments having similar lengths may appear to be a single fragment. This is known as fragment collapsing (Gillett *et al.* 1995). Clone chimerism is a problem associated with all mapping protocols. Chimerism is especially common with certain cloning vectors, such as yeast artificial chromosomes. Finally, there are typically gaps in library coverage that create contig breaks. The relative order and orientation of contigs cannot be deduced without additional data (Greenberg and Istrail 1995).

Many algorithmic approaches have been developed to recover the native order of clones with noisy data. Algorithms for restriction enzyme fingerprinting data are given in Coulson *et al.* (1986), Carrano *et al.* (1989), Stallings *et al.* (1990), and Gillett *et al.* (1995). Algorithms for oligonucleotide hybridization data are given in Cuticchia *et al.* (1992), Fu *et al.* (1992), and Mayraz and Shamir (1999). Algorithms for STS content and clone hybridization data are given in Cuticchia *et al.* (1992), Mott *et al.* (1993), Wang *et al.* (1994), Alizadeh *et al.* (1995), Greenberg and Istrail (1995), Nadkarni *et al.* (1996), Bhandarkar and Machaka (1997), Christoff *et al.* (1997), Jain and Myers (1997), Christoff and Kececioglu (1999), Kececioglu *et al.* (2000), and Tsai and Kao (2000). An algorithm that uses both clone hybridization data

Corresponding author: Suchendra M. Bhandarkar, Department of Computer Science, The University of Georgia, 415 Boyd Graduate Studies Research Ctr., Athens, GA 30602-7404.
E-mail: suchi@cs.uga.edu

and restriction enzyme fingerprinting is described in Sasinowska and Sasinowski (1999).

A number of freely available software applications for creating contig maps have been developed over the past decade. Applications for restriction digestion data include CONTIG9 (Sulston *et al.* 1988), GRAM (Soderlund and McGarvan 1993), and FPC (Soderlund *et al.* 1997). A number of applications have been developed for creating STS marker maps. These include SEGMAP (Green and Green 1991; Magness *et al.* 1994), ContigMaker (Suyama 1993), CONTIGMAKER (Daly *et al.* 1994), SAM (Soderlund and Dunham 1995), and Contig Explorer (Nadkarni *et al.* 1996). Applications for creating contig maps using clone hybridization data include Probeorder, Costig, and Bar (Mott *et al.* 1993), and ODS (Cuticchia *et al.* 1993), which was also designed for oligonucleotide fingerprinting data.

An effort is currently under way to create high-resolution cosmid-based contig maps of the fungus *Neurospora crassa* (http://gene.genetics.uga.edu; Aign *et al.* 2001; Bhandarkar *et al.* 2001; Kelkar *et al.* 2001). The approach that is being taken uses hybridization to detect overlaps between 40-kb cosmids. A hybridization-based protocol is advantageous in that a high degree of parallelism can be achieved. DNA from thousands of clones can be fixed to the same nylon filter and simultaneously probed for hybridization with a clone. If the clones have been assigned to chromosomes, then the probings can be done with a mixture of clones, one from each chromosome, to achieve a speedup proportional to the number of chromosomes. Additionally, robotics systems can be extensively utilized in the process to increase laboratory throughput (Arnold and Cushion 1999).

Although a hybridization-based approach is relatively inexpensive due to the economy of scale that can be realized, the reliability of maps created using only hybridization data is considered to be less than that for maps constructed using STS content data (Sasinowska and Sasinowski 1999). *N. crassa* has a long history of genetic study and, consequently, rich genetic maps are available for this organism (Perkins 2000). An effort is being made in the Neurospora mapping project to anchor clones to these genetic maps through direct genetic complementation, hybridization to clones known to complement mapped mutations, and BLAST search with available clone sequence data such as cosmid end-sequence data (Kelkar *et al.* 2001). This genetic data will help researchers to construct more reliable maps. First, it will allow anchored contigs to be placed in the correct position and orientation relative to one another. Second, it will help researchers detect and correct false joins. These occur when multiple contigs appear to be a single contig due to false-positive errors and chimerism.

Of the existing software, ODS (Cuticchia *et al.* 1993), Probeorder, Costig, and Bar (Mott *et al.* 1993) are the most suitable in that they were designed for building contigs of small eukaryotic genomes using clone-clone hybridization data. The program ODS was used in the construction of the *A. nidulans* physical maps (Prade *et al.* 1997). The programs Probeorder, Costig, and Barwere used in the mapping of the *Schizosaccharomyces pombe* genome (Mott *et al.* 1993). Neither of these applications contains any direct support for integrating physical and genetic maps, however. In this article we report an updated version of the ODS package called ODS2. It uses a modified version of the algorithm featured in ODS that incorporates genetic mapping data as well as clone-clone hybridization data. ODS2 has a graphical user interface and was implemented in Java, so it runs on multiple platforms (*e.g.*, UNIX and MS Windows). It also features a graphical tool for displaying and editing maps. We discuss these features and illustrate the use of the software in the construction of a preliminary map of *N. crassa* linkage group VI.

MATERIALS AND METHODS

Cosmid libraries used to construct the physical maps in Figures 5 and 6 are described in Kelkar *et al.* (2001). Physical mapping data as shown in Figure 6 were generated by DNA hybridization described in Kelkar *et al.* (2001). Assignments of markers to physical and genetic maps was achieved by complementation, hybridization, and cosmid end sequencing as described in Kelkar *et al.* (2001).

ALGORITHM

Description of clone ordering algorithm: The clone ordering algorithm used by ODS2 is a modification of that used in ODS (Cuticchia *et al.* 1993). The algorithms used by ODS are based on Hamming distance and simulated annealing. Briefly, the Hamming distance between each pair of clones is computed. The Hamming distance between two clones is the number of probes hybridizing to one clone, but not to both. Simulated annealing is then used to search for a permutation of clones that minimizes the sum of Hamming distances between each pair of adjacent clones.

The algorithm used by ODS2 incorporates the following modifications and enhancements over the one used by ODS. First, the Hamming distances between probes are computed and the probes are ordered instead of clones. Let $C = c_1, c_2, \ldots, c_{|C|}$ denote the set of clones. Let $P = p_1, p_2, \ldots, p_{|P|}$ denote the set of probes. Let $P^\pi = p^\pi_2, p^\pi_2, \ldots, p^\pi_{|P|}$ denote an ordering (or permutation) of the probes. Let D denote a binary matrix of size $|C| \times |P|$, where $D_{i,j}$ is 1 if clone c_i is believed to contain probe p_j based on experimental data, or 0 otherwise. Let D^π denote a matrix derived from D by permuting the columns to correspond to the probe ordering P^π. The Hamming-distance traveling salesman objective function is given by

$$F(P^\pi) = \sum_{i=1}^{|P|-1} \sum_{j=1}^{|C|} D^\pi_{j,i} \oplus D^\pi_{j,i+1}, \qquad (1)$$

where $\oplus$ is the Boolean exclusive or operation. This modification was made to decrease the runtime of the algorithm, as the set of probes is a subset of the set of clones. After ordering the probes, the clones are placed within the probe order as follows. For a given clone, the longest contiguous sequence of probes that hybridize with the clone is found, if such a sequence exists. The clone is placed in the map such that it spans across these probes. This approach of ordering probes and then fitting clones to the probe order is also described in Mott *et al.* (1993). If more than one such sequence is present, then the clone cannot be placed unambiguously in the map. Such clones are randomly placed in one of the possible positions. These clones with ambiguous location are indicated to the user by color coding. The user may then choose to eliminate these clones from the map or use genetic complementation data, if available, to correctly place the clones. Of course, clones that do not hybridize to any probes cannot be placed in the map.

The second modification to the algorithm is the use of the microcanonical annealing search algorithm (Creutz 1983) instead of simulated annealing. Microcanonical annealing was found to achieve levels of optimization as good as simulated annealing and to do so an order of magnitude faster (Bhandarkar and Machaka 1997). This algorithm, as adapted for clone ordering, is shown in Figure 1.

The third modification is a weighted penalty value ϕ that is added to the sum of Hamming distances $F(P^\pi)$. This penalty is related to the number of misplaced anchored clones. When a given permutation of probes is being evaluated by the algorithm, the subset of clones that are anchored to the genetic map are placed within the probe ordering as described above. Let $A^\pi = <a_1^\pi, a_2^\pi, \ldots, a_{|A|}^\pi>$ denote a permutation of anchored clones. Let $\text{pos}(a_i^\pi)$ return the position in the genetic map of the marker to which a_i^π is anchored. One of the penalty functions ϕ_1 used by ODS2 is given by

$$\phi_1 = \alpha \sum_{i=0}^{|A|-1} |\text{pos}(a_i^\pi) - \text{pos}(a_{i+1}^\pi)|. \quad (2)$$

The variable α is a scaling factor that can be adjusted by the user. When α is set to a higher value, more emphasis is placed on genetic complementation data. Note that this penalty is a minimum for a given α when the order of markers implied by A^π is the same as the order of markers in the genetic map.

The second penalty function ϕ_2 used by ODS2 does not require the set of markers to be completely ordered. For $i < j$, let

$$R(a_i^\pi, a_j^\pi) = \begin{cases} 1, & \text{if } a_i^\pi \text{ is to the right of } a_j^\pi \text{ in the genetic map} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Then ϕ_2 is computed as

```
1. for i ← 1 to N
2.         for j ← 1 to N
3.                 E[i][j] ← E_max
4. V ← initial value of objective function
5. while not finished begin
6.         for i ← 1 to MAXCOUNT begin
7.                 Randomly choose p and q such that 1 ≤ p < q ≤ N
8.                 Reverse block of probes between and including p and q
9.                 ΔV ← change in objective function
10.                Accept permutation if one of the following is true
11.                        (i) ΔV < 0
12.                        (ii) 0 ≤ ΔV ≤ E[p][q]
13.                if permutation is accepted begin
14.                        E[p][q] = E[q][p] = E[p][q] - ΔV
15.                        V ← V - ΔV
16.                endif else
17.                        Restore previous probe order
18.        endfor
19.        for i ← 1 to N
20.                for j ← 1 to N
21.                        E[i][j] = E[i][j] × factor
22.        if V is unchanged for K iterations halt
23. endwhile
```

Figure 1.—The microcanonical annealing algorithm search algorithm used by ODS2. N is the number of probes. Parameters that were used are as follows: $E_{max} = 0.5$; factor = 0.5; $K = 3$; MAXCOUNT = $100 \times N$.

$$\phi_2 = \alpha \sum_{1 \le i < j \le |A|} R(a_i^\pi, a_j^\pi). \quad (4)$$

The penalty function ϕ_2 penalizes each pair of markers in A^π whose order relative to each other is incorrect. Since ϕ_2 exploits only the pairwise ordinal information between markers it can handle partially ordered marker information. As before, α is a scaling factor that can be adjusted by the user. Note that ϕ_2 has a minimum value when the order of markers implied by A^π is the same as the order of markers in the genetic map.

The third objective function ϕ_3 is described in Jain and Myers (1997) and can be expressed as

$$\phi_3 = \alpha(|A| - \text{lis}(A^\pi)), \quad (5)$$

where $\text{lis}(A^\pi)$ is the length of the longest increasing subsequence in A^π. As before, α is a scaling factor that can be set by the user.

Evaluation of the algorithm on simulated data: The probe ordering algorithm in ODS2 was tested with the three penalty functions ϕ_1, ϕ_2, and ϕ_3 described above on simulated data modeled after data from the *A. nidulans* physical mapping project (Prade *et al.* 1997). A 4-Mb chromosome was modeled and 40-kb clones were generated by a Poisson process that produced a chromosomal coverage of 5. Nonoverlapping probes were randomly

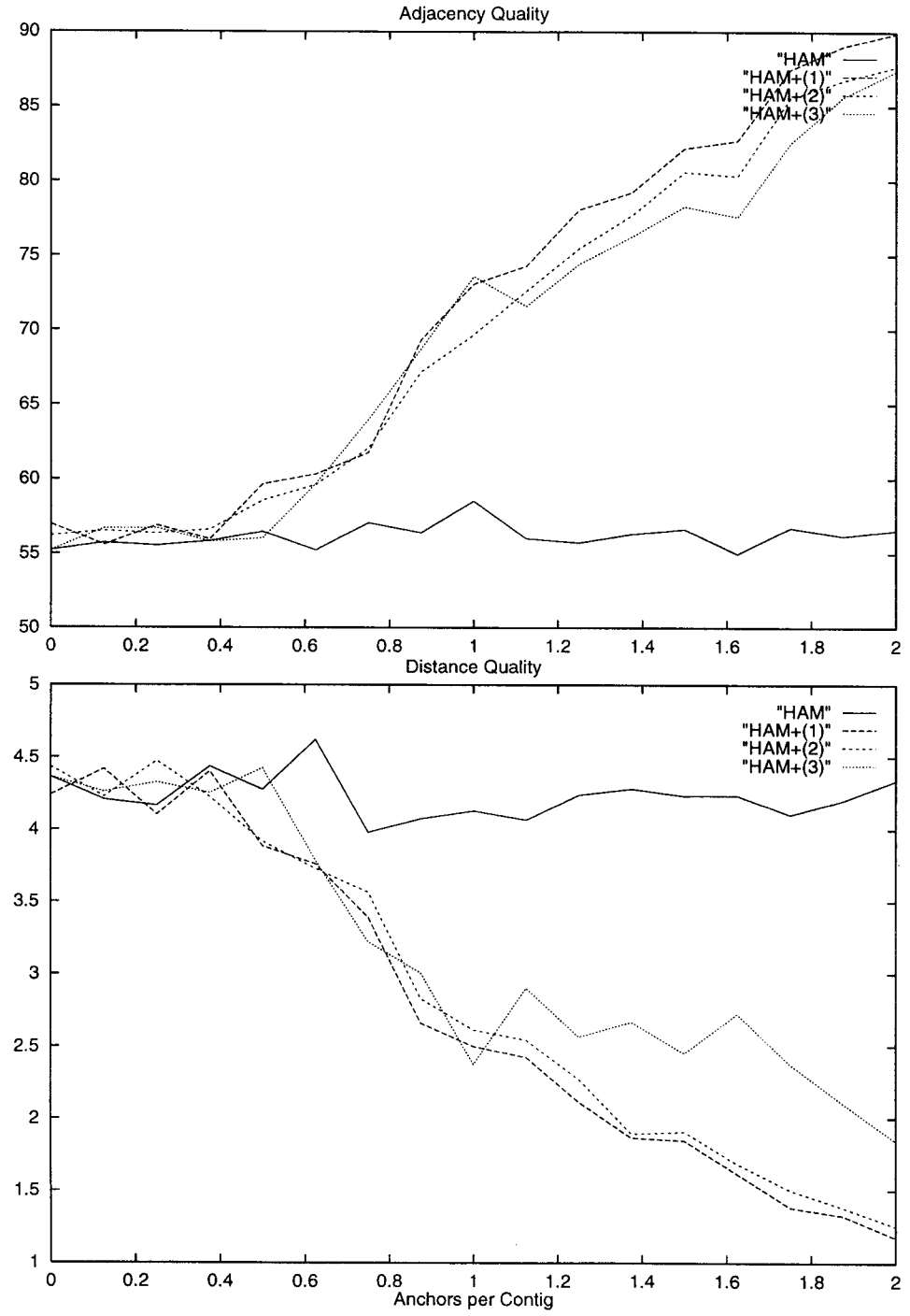

FIGURE 2.—Adjacency and distance quality *vs.* the average number of anchors per contig. Probes were ordered using the microcanonical annealing algorithm and the Hamming-distance traveling salesman objective function $F(P^{\pi})$ by itself (HAM) and augmented with each of the penalty functions ϕ_1[HAM + (1)], ϕ_2 [HAM + (2)], and ϕ_3 [HAM + (3)]. Each point represents the mean of 50 separate problem incidences on simulated data.

selected without replacement from the clones to give a probe coverage of 0.8. A probe-clone incidence matrix was generated by scoring all overlapping clone and probe pairs as 1 and all nonoverlapping pairs as 0. False-positive and false-negative errors were introduced at random into the matrix at the rates of 1 and 20%, respectively. The simulated data were prefiltered for removal of potential false-positive errors using an algorithm proposed by MOTT *et al.* (1993).

During preliminary simulations it became apparent that improvements in the probe orderings were most strongly correlated with the distribution of the anchored markers. Two probes are connected if they are believed to be incident to the same clone. A contig of probes is a set of probes where each probe in that contig is connected to at least one other probe in the contig, and no probes are connected to a probe in any other contig. It was shown that the greatest improvements in map quality came when there were a sufficient number of anchored markers distributed uniformly among the probe contigs.

Two different measures of similarity between permu-

TABLE 1

Values for selected points in Figure 2

Anchors per contig	Objective function			
	$F(P^\pi)$	$F(P^\pi) + \phi_1$	$F(P^\pi) + \phi_2$	$F(P^\pi) + \phi_3$
	Average adjacency quality (%)			
0.0	55 ± 14	57 ± 14	56 ± 14	55 ± 14
0.5	56 ± 16	60 ± 13	59 ± 13	56 ± 13
1.0	59 ± 15	73 ± 15	70 ± 15	74 ± 13
1.5	57 ± 13	82 ± 15	81 ± 13	78 ± 17
2.0	57 ± 16	89 ± 11	88 ± 12	88 ± 15
	Average distance quality			
0.0	4.37 ± 1.96	4.24 ± 2.65	4.43 ± 2.67	4.37 ± 1.94
0.5	4.28 ± 2.60	3.88 ± 2.21	3.91 ± 2.29	4.42 ± 2.50
1.0	4.13 ± 2.45	2.50 ± 1.72	2.61 ± 1.55	2.37 ± 1.37
1.5	4.23 ± 2.18	1.84 ± 1.51	1.90 ± 1.41	2.45 ± 1.53
2.0	4.34 ± 2.22	1.17 ± 1.16	1.25 ± 1.25	1.84 ± 1.80

Average adjacency and distance qualities for the Hamming-distance traveling salesman objective function $F(P^\pi)$ by itself and augmented with the penalty functions ϕ_1, ϕ_2, and ϕ_3 on simulated data with the ±95% error bounds (*i.e.*, confidence intervals) are shown.

tations were used to evaluate the probe orderings generated from simulated data. These are the adjacency quality (AQ) and the distance quality (DQ; Greenberg and Istrail 1995). The adjacency quality is the fraction of adjacencies in the computed ordering of probes that exist in the true ordering. The distance quality is the average number of positions in the computed ordering separating two probes that are adjacent in the true ordering. For adjacency quality the ideal value is 100 whereas for distance quality the ideal value is 1. In Figure 2, simulated data were ordered using the microcanonical annealing algorithm and the Hamming-distance traveling salesman objective function $F(P^\pi)$ by itself, and $F(P^\pi)$ augmented with each of the three penalty functions ϕ_1, ϕ_2, and ϕ_3. Each point in the graphs represents the mean of 50 separate runs on different data sets. These graphs show that as the average number of anchored probes per contig increases, the quality of the resulting map improves when the penalty functions are used. Mean values with 95% error bounds (*i.e.*, confidence intervals) for some of the points in these graphs are shown in Table 1.

As mentioned previously, penalty function ϕ_2 can be used when the set of anchored markers is not totally ordered. To increase the total number of anchored probes, it may be desirable to pool together markers that have been anchored to different maps. In Figure 3, we simulated the pooling together of different sets of anchored markers. The cardinality of each separate set of anchored markers is equal to 0.7 times the number of probe contigs. Figure 3 shows the impact of combining one, two, and three separate sets of anchored markers on the adjacency quality and the distance quality of the resulting probe ordering. Each bar shows the mean of 50 separate runs on different simulated data sets. These plots show that combining anchored probe data from multiple independent sources using penalty function ϕ_2 can improve the accuracy of the computed order of probes.

Evaluation of the algorithm on real data: The probe ordering algorithm in ODS2 was also evaluated in terms of run time and map quality using real data from *A. nidulans* chromosomes I, VII, and VIII (Prade *et al.* 1997). Extensive genetic maps are available for this organism. Ideally, the physical maps should contain as few contigs as possible and agree with the genetic map. Due to hybridization experimental error, the contig map produced by the ordering algorithm may not agree with the genetic map. We used the following method to measure the correlation between the contig and genetic maps. By removing all nonanchored clones from a physi-

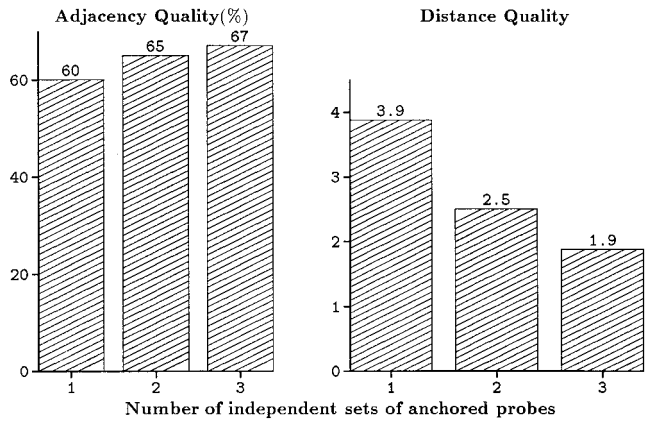

Figure 3.—Adjacency and distance quality as a function of the number of independent sets from which the anchored probe data were derived. Each bar represents the average of 50 runs on distinct simulated data sets.

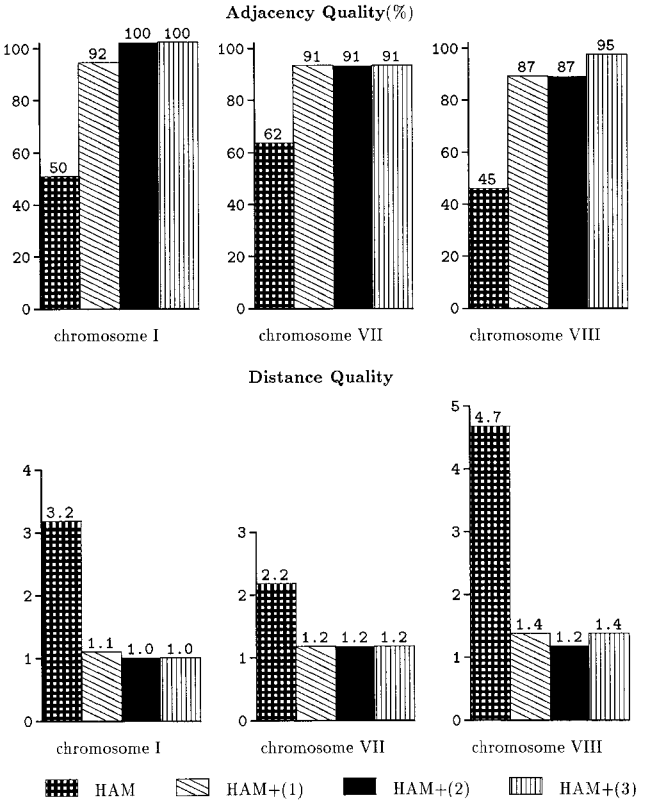

FIGURE 4.—Results from ordering *A. nidulans* data using the Hamming-distance traveling salesman objective function with and without penalty functions ϕ_1 (1), ϕ_2 (2), and ϕ_3 (3). The bar plot represents the adjacency or distance quality of genetic markers in the physical map and represents the average of five runs with $\alpha = 1.0$.

cal map, we are left with a permutation of anchored probes and clones and, hence, an ordering of the markers. The AQ and DQ measurements were then determined for this ordering of markers implied by the ordering of the probes. Also, the Hamming-distance traveling salesman objective function value $F(P^{\pi})$ was determined for the computed ordering of probes to see if the incorporation of the penalty functions led to an increase in its value. This gave an indication of the consistency of the probe orderings with the clone-probe incidence data.

Figure 4 shows that when the penalty functions were incorporated, the resulting probe orderings were more consistent with the genetic maps. Since microcanonical annealing is a stochastic optimization algorithm, the data in Figure 4 represent the average of five different runs. Table 2 shows the observed increase in the Hamming-distance traveling salesman objective function value $F(P^{\pi})$ when the penalty functions were incorporated. Each value in Table 2 represents the average of five different runs. In most cases, incorporating anchored marker data using the penalty functions leads to a slight increase ($<1\%$) in the final value of $F(P^{\pi})$. In the case of chromosome I, using penalty function ϕ_2 caused the value of $F(P^{\pi})$ to decrease. In the cases of chromosomes VII and VIII the incorporation of penalty function ϕ_3 resulted in more substantial increases in the value of $F(P^{\pi})$ (Table 2).

Benchmarks of the program ODS2 were run on a Sun Enterprise 250 computer with a 300 MHz UltraSparcII CPU and 512 MB main memory running the Solaris 7 operating system on data from chromosome VIII of *A. nidulans* using penalty function ϕ_1. Results using a number of different values for α are reported in Table 3. As microcanonical annealing is a stochastic algorithm, benchmarks were run 25 times for each α value. Average values over the 25 runs are reported in Table 3.

In general, run times for these data are reasonably short. It can be seen that, in general, as more weight is placed on the penalty function, both the number of contig breaks and the percentage of marker adjacencies recovered increase. When α was equal to 0.01, there were, on average, 16.4 contigs and 55% of marker adjacencies were recovered. When α was equal to 100.0, there were, on average, 20.1 contigs and 100% of marker adjacencies were recovered. This indicates that the algorithm is creating contig breaks to align the physical map with the genetic map. The program ODS2 also supports the manual alignment of physical and genetic maps. This is described in the following section.

IMPLEMENTATION

ODS2 is a fully graphical application implemented in Java. It features a graphical user interface and will run on any platform that supports Java 2. It has been tested on Sun hardware running the Solaris 7 operating system and on PC hardware running Microsoft Windows NT. In addition to providing tools for creating maps, ODS2 also provides features for data management. The software can store data in a file or in a database on the same host or a remote host. Data can be added incrementally to either of these repositories using the application at any time during the life of a project. Data are entered into the repositories via five types of text files. The first three files contain data that are required, and the last two contain optional data. The first type of required data indicates to which chromosome(s) or linkage group(s) each clone in a set of clones belongs. Note that a clone may belong to more than one chromosome or linkage group. This situation may arise if clones are mapped to chromosomes by hybridization. Common repetitive sequences may cause a clone to hybridize to more than one chromosome. The second type of required data involves probe names for a chromosome or linkage group. The third type of required data indicates which probes have hybridized to each clone in a set of clones. The first optional data type is an encoding of a genetic map. The second optional data type lists genetic markers and indicates which clones contain

TABLE 2

Change in final value of the Hamming-distance traveling salesman function of *A. nidulans* data after ordering with the penalty functions *vs.* without

Objective function	Chromosome I	Chromosome VII	Chromosome VIII
HAM + (1)	+0.29%	+0.44%	+0.80%
HAM + (2)	−0.03%	+0.40%	+0.30%
HAM + (3)	+0.90%	+1.12%	+1.90%

Data were ordered with the Hamming-distance objective function alone and in combination with one of the penalty functions. The value of the Hamming-distance traveling salesman function for the data ordered using each penalty function, minus the value for data ordered without the penalties, is shown. Average values over five runs are presented with $\alpha = 1.0$.

them. The precise format of each of these file types is specified in the documentation that comes with the software.

The program ODS2 can be used with either a relational database or a special file for managing all the data for a given project. A typical session with the software may begin by opening an existing project. To do this, the user would load all data from the project file or database into memory. All software functions can be carried out by selecting from menus or activating other graphical components. The user may then import new data from one or more of the text files described above. ODS2 has a menu option for configuring a database. The user specifies a database account name, password, host name, and port. The application will create the necessary tables if they do not already exist. ODS2 has been tested with Oracle 8 (http://www.oracle.com) and MySQL (http://www.mysql.com) database management systems on Sun hardware running the Solaris 7 operating system. The application should theoretically work with any relational database management system that is SQL92 compliant and for which there is a Java JDBC driver available.

After opening a project, the user may then select one or more chromosomes and build a map for them. The user can elect whether or not to use any of the penalty functions described previously and can adjust the weight (α) placed on this function. When the selected map(s) are generated, each is displayed graphically in a new window. This window contains a menu that lets users edit the associated map. This editing tool is shown in Figure 5, which displays a map of *N. crassa* linkage group VI. On the far left side of the screen, a colored bar indicates the depth of coverage at the corresponding

TABLE 3

Analysis of the ODS2 algorithm with penalty function ϕ_1 on data from *A. nidulans* chromosome VIII

α	Run time (sec)	No. of contigs	Recovered marker adjacencies (%)
0.01	11.0	16.4	55
0.1	10.4	15.7	57
1.0	12.8	18.6	92
10.0	11.6	19.6	100
100.0	13.3	20.1	100

Benchmarks were performed on a Sun E250 computer, with a 300 MHz UltraSparcII CPU and 512 MB main memory, running the Solaris 7 operating system. The data contain 1273 clones, 118 probes, and 12 anchored clones. The algorithm was run 25 times for each value of α. Average values are reported.

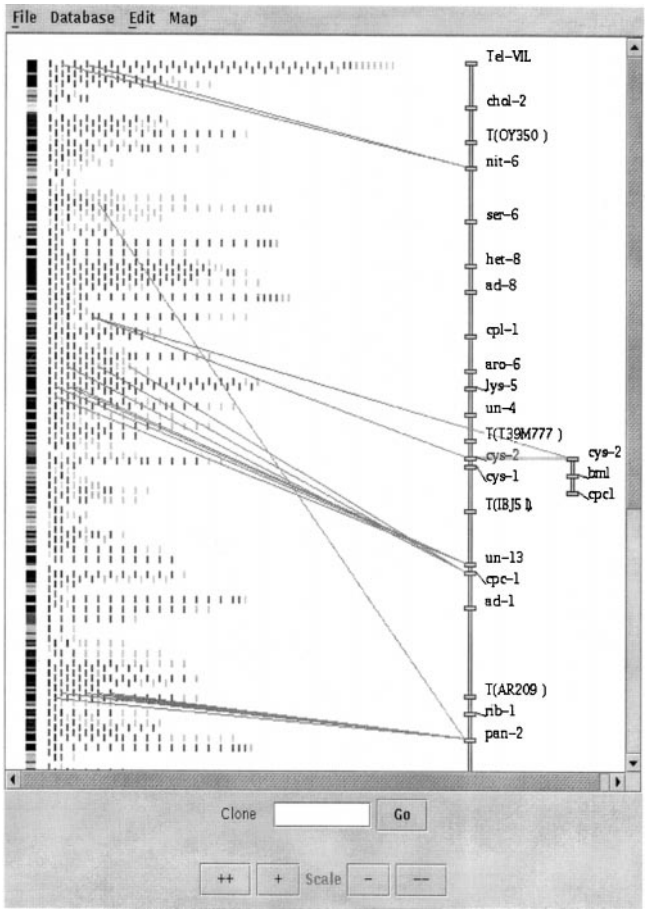

FIGURE 5.—Screenshot of the ODS2 graphical map viewing/editing tool displaying part of an integrated physical and genetic map of *N. crassa* linkage group VI.

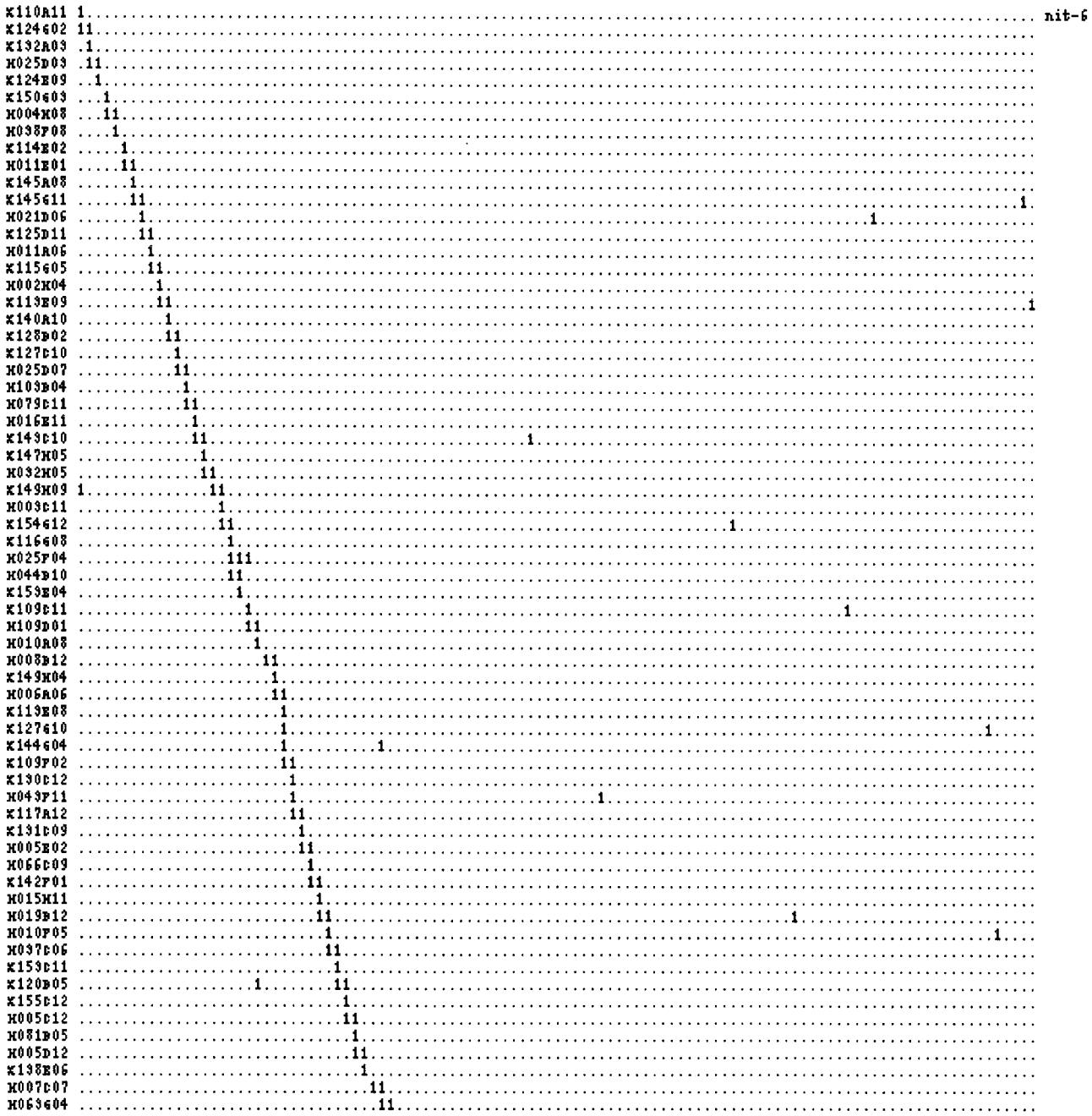

FIGURE 6.—A minimal tiling contig map of *N. crassa* linkage group VI. Clone names are in the left column. If a clone contains a marker, the name of the marker is displayed in the far right column. Hybridization data are shown as a matrix of binary values. Columns correspond to probes. A "1" indicates that a clone and probe hybridize and a "." indicates a clone and probe do not hybridize.

position on the map. The darker the color, the greater the coverage at that position. Contig breaks (*i.e.*, regions of 0 coverage) are indicated by red bands (these do not show up in the grayscale image of this figure). To the right of the coverage indicator are the clone contigs. The clones are color coded as follows. Black clones were used as probes. Blue clones can be unambiguously placed within the probe ordering. Gold clones cannot be placed unambiguously within the probe ordering (again, these colors are not evident in the grayscale image). The scale of the displayed maps can be adjusted as needed.

The genetic map is displayed to the right of the contigs. Clones anchored to genetic markers are indicated by gray lines that terminate at one end on the clones and at the other end on the associated markers. If the contig map is in agreement with the genetic map, then none of these lines should cross each other. After the

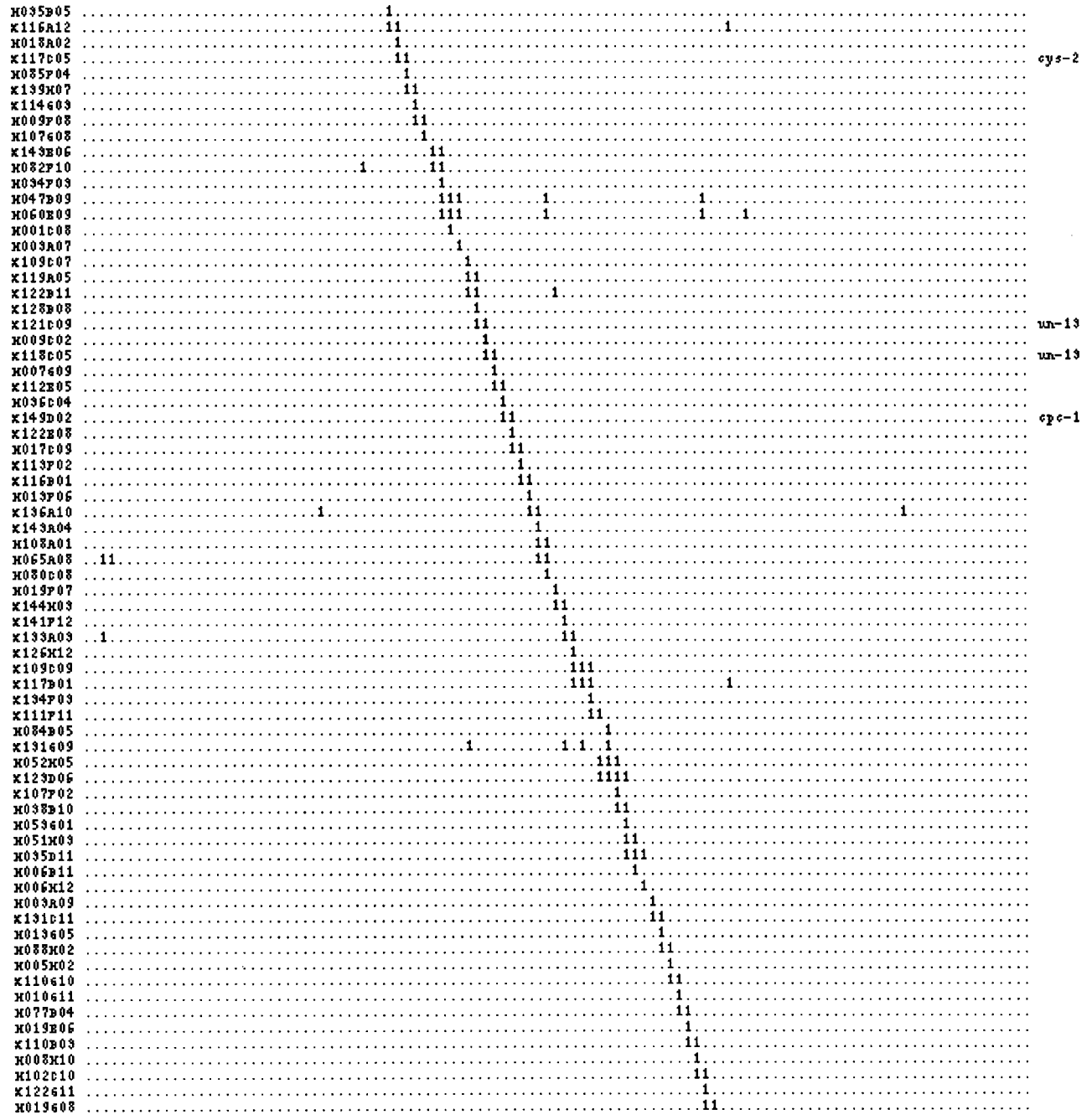

FIGURE 6.—*Continued.*

software orders the clones, several types of corrections to the map may be necessary. It may be necessary to reposition the contigs. Also, false contigs that are detected when clones within the same contig are anchored to different regions of the genetic map may need to be broken. An example of this situation can be seen in Figure 5, where a contig must be broken so that the uppermost subcontig containing the marker pan-2 (bottom of genetic map) can be moved next to the other contig containing this marker.

Several types of map editing operations are supported. First, blocks of clones can be selected and reversed. A block of clones can also be moved to a different location on the map. This is accomplished by selecting a set of clones by dragging the mouse over them, cutting these clones out of the map, and pasting them to a different location. In this way entire contigs can be moved or possibly false contigs broken up and separated. The depth indicator to the left of the clones is useful for moving and reversing clone blocks. For moving an entire contig, the user selects clones between two red bands on the indicator. For breaking a potential false contig, the user would want to make the break in a region where there is little coverage; *i.e.*, the user

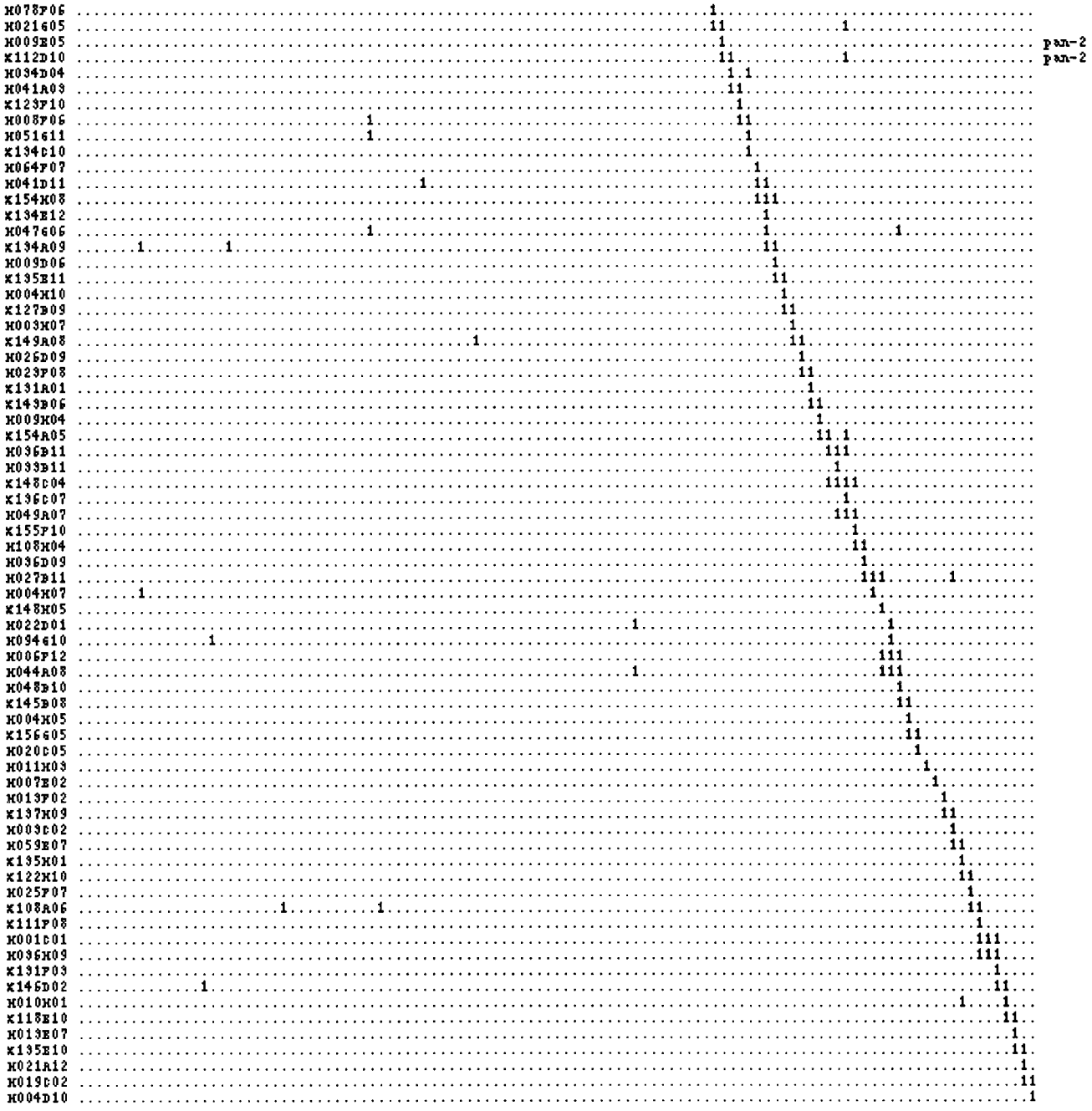

FIGURE 6.—*Continued.*

would move clones between two light-colored bands or a light-colored band and a red band. Light-colored bands correspond to regions of the map where there are relatively few observed hybridizations to link subcontigs. Random false-positive hybridization could cause contigs to be falsely joined, and the junction between contigs would likely have a corresponding light-colored band due to a lack of additional true hybridizations.

Several other types of edits are also supported. Clones may be selected and deleted from the map. This option would perhaps be used to delete ambiguously placed clones individually. There is a menu option for deleting all ambiguously placed nonanchored clones at one time as well. The software will also generate a minimum tiling map. It does this by removing all clones from the map except for a minimal set needed to cover the chromosome. Clones are preferentially kept in the map if they are a probe, or they can be unambiguously placed in the map, or they are anchored to the genetic map. Ambiguously placed clones can be automatically moved to alternative locations using a menu option.

After a map is built, it may be saved in a file on disk or saved to a database. Previously built maps can be loaded from either of these sources at a later time. Maps

can also be exported into a text format. Such a map is shown in Figure 6. This map is a preliminary minimum tiling of *N. crassa* linkage group VI (the data collection is ~75% complete as of date). The map in Figure 6 was generated using a weight of $\alpha = 10$ for the genetic mapping data. The map underwent additional manual editing to bring it into alignment with the genetic map. A preliminary full version of the map is available for viewing on the web at http://gene.genetics.uga.edu/ncrassa6.html.

DISCUSSION

Although a number of good software applications have been developed over the years to support the construction of contig maps, none of these is ideally suited to support the mapping protocol being taken in the *N. crassa* project. ODS (Cuticchia *et al.* 1993), Probe, Costig, and Barr (Mott *et al.* 1993) were designed for creating maps using clone-clone hybridization data, but do not contain any features for integrating genetic mapping data into the contig maps. It would be possible to use one of the applications for creating STS marker maps, such as SEGMAP (Green and Green 1991; Magness *et al.* 1994), as some of them also build contigs as well. However, SEGMAP was designed for creating yeast artificial chromosome-based maps of larger genomes. For instance, one of the inputs for the program is a file containing the chromosomal banding patterns. Thus, SEGMAP would not be suitable for fungal chromosomes. Most of the existing mapping applications are tied to a particular computer platform or database management software. For instance, Contig Explorer (Nadkarni *et al.* 1996) runs on Macintosh clients and uses a UNIX server for data storage.

Although ODS2 was created for the *N. crassa* mapping project, it would be a good candidate for use in other mapping projects as well. ODS2 was designed, in particular, for genomes with good genetic maps. However, the application has other features that could make it a viable tool even for genomes without genetic maps. First, as it is Java based, it runs on virtually all modern platforms. Second, it gives the users the flexibility of storing data in local files or a central database. Third, it has a completely graphical user interface. Many of the menu operations, such as "cut" and "paste," are common to other modern applications. Fourth, the clone ordering approach based on Hamming distances has been proven in other mapping projects (Prade *et al.* 1997; Enkerli *et al.* 2000). Fifth, full maps or minimal tiling maps can be built. We are making ODS2 freely available for noncommercial use. It can be obtained by contacting the authors.

In conclusion, the integration of multiple data and information sources is critical in genomics projects to enhance the accuracy and reliability of the final product or conclusion. It is clear that more research and more software tools are needed in this area. Our future work will focus on the integration of the maximum-likelihood-based physical mapping objective function (Bhandarkar *et al.* 2001) with clone data that are anchored to genetic markers on the genetic map. This would involve augmenting the maximum-likelihood objective function with ordinal information in the form of a prior distribution derived from the anchored clones. We are also investigating normalization of transcription profiling data through integration with views of the physical map.

This research was supported in part by an NRICGP grant from the U.S. Department of Agriculture and in part by a Microbial Genetics Grant MCB-9630910 from the National Science Foundation.

LITERATURE CITED

Aign, V., U. Schulte and J. D. Hoheisel, 2001 Hybridization mapping of *Neurospora crassa* linkage groups II and V. Genetics **157:** 1015–1020.

Alizadeh, F., R. M. Karp, D. K. Weisser and G. Zweig, 1995 Physical mapping of chromosomes using unique probes. J. Comp. Biol. **2:** 159–184.

Arnold, J., and M. T. Cushion, 1999 Constructing a physical map of the *Pneumosystis* genome. J. Eukaryot. Microbiol. **44:** 8S.

Bhandarkar, S. M., and S. A. Machaka, 1997 Chromosome reconstruction from physical maps using a cluster of workstations. J. Supercomput. **11:** 61–86.

Bhandarkar, S. M., S. A. Machaka, S. S. Shete and R. N. Kota, 2001 Parallel computation of a maximum likelihood estimator of a physical map. Genetics **157:** 1021–1043.

Carrano, A. V., J. Lamerdin, L. K. Ashworth, B. Watkins, E. Bascomb *et al.*, 1989 A high-resolution, fluorescence-based, semiautomated method for DNA fingerprinting. Genomics **4:** 129–136.

Christoff, T., and J. Kececioglu, 1999 Computing physical maps of chromosomes with nonoverlapping probes by branch and cut. Proceedings of the 3rd ACM Conference on Computational Molecular Biology, Lyon, France, pp. 115–123.

Christoff, T., M. Jünger, J. Kececioglu, P. Mutzel and G. Reinelt, 1997 A branch-and-cut approach to physical mapping of chromosomes by unique end-probes. J. Comp. Biol. **4:** 433–447.

Chumakov, I. M., P. Rigault, I. Le Gall, C. Bellanne-Chantelot, A. Billault *et al.*, 1995 A YAC contig map of the human genome. Nature **377** (Suppl): 175–297.

Coulson, A., J. Sulston, S. Brenner and J. Karn, 1986 Toward a physical map of the genome of the nematode *Caenorhabditis elegans*. Proc. Natl. Acad. Sci. USA **83:** 7821–7825.

Creutz, M., 1983 Microcanonical Monte Carlo simulation. Phys. Rev. Lett. **50:** 1411–1414.

Cuticchia, A. J., J. Arnold and E. Timberlake, 1992 The use of simulated annealing in chromosome reconstruction experiments based on binary scoring. Genetics **132:** 591–601.

Cuticchia, A. J., J. Arnold and W. E. Timberlake, 1993 ODS (ordering DNA sequences): a physical mapping algorithm based on simulated annealing. Comput. Appl. Biosci. **9:** 215–219.

Daly, M. J., M. P. Reeve, A. Kaufman, J. Orlin and E. S. Lander, 1994 CONTIGMAKER: software for physical map contig assembly. Cold Spring Harbor Meeting on Genome Mapping and Sequencing, Cold Spring Harbor, NY, p. 210.

Enkerli, J., H. Reed, A. Briley, G. Bhatt and S. F. Covert, 2000 Physical map of a conditionally dispensable chromosome in *Nectria haematococca* MP VI and location of chromosome breakpoints. Genetics **155:** 1083–1094.

Fu, Y., W. E. Timberlake and J. Arnold, 1992 On the design of genome mapping experiments using short synthetic oligonucleotides. Biometrics **48:** 337–359.

Gillett, W., J. Daues, L. Hanks and R. Capra, 1995 Fragment collapsing and splitting while assembling high-resolution restriction maps. J. Comp. Biol. **2:** 185–205.

Green, E. D., and P. Green, 1991 Sequence-tagged site (STS) con-

tent mapping of human chromosomes: theoretical considerations and early experiences. PCR Methods Appl. **1:** 77–90.

Green, E. D., and M. V. Olson, 1990 Systematic screening of yeast artificial-chromosome libraries by use of the polymerase chain reaction. Proc. Natl. Acad. Sci. USA **87:** 1213–1217.

Greenberg, D. S., and S. Istrail, 1995 Physical mapping by STS hybridization: algorithmic strategies and the challenge of software evaluation. J. Comp. Biol. **2:** 219–273.

Jain, M., and E. W. Myers, 1997 Algorithms for computing and integrating physical maps using unique probes. J. Comp. Biol. **4:** 449–466.

Kececioglu, J., S. Shete and J. Arnold, 2000 Reconstructing order and distance in physical maps using nonoverlapping probes. Proceedings of the 4th ACM Conference on Computational Molecular Biology, Tokyo, Japan, pp. 183–192.

Kelkar, H. S., J. Griffith. M. E. Case, S. F. Covert, R. D. Hall *et al.*, 2001 The Neurospora crassa genome: cosmid libraries sorted by chromosome. Genetics **157:** 979–990.

Lehrach, H., R. Drmanac, J. Hoheisel, Z. Larin, G. Lennon *et al.*, 1990 Hybridization fingerprinting in genome mapping and sequencing, pp. 39–82 in *Genetic and Physical Mapping*, Vol. 1, edited by K. E. Davies and S. M. Tilghman. Cold Spring Harbor Press, Plainview, NY.

Magness, C., Y. Xu and P. Green, 1994 SEGMAP: an interactive computer program for generating YAC-based STS-content maps. First International Workshop on Human Chromosome 7 Mapping, Cytogenetics and Cell Genetics (1-2), Marburg, Germany, p. 63.

Mayraz, G., and S. Shamir, 1999 Construction of physical maps from oligonucleotide fingerprints data. Proceedings of the 3rd ACM Conference on Computational Molecular Biology, Lyon, France, pp. 268–277.

McPherson, J. D., 1997 Sequence ready—or not? Genome Res. **7:** 1111–1113.

Mott, R., A. Grigoriev, E. Maier, J. Hoheisel and H. Lehrach, 1993 Algorithms and software tools for ordering clone libraries: application to mapping of the genome of *Schizosaccharomyces pombe*. Nucleic Acids Res. **21:** 1965–1974.

Nadkarni, P. M., A. Banks, K. Montgomery, J. Leblanc-Stracewski, P. Miller *et al.*, 1996 CONTIG EXPLORER: interactive marker-content map assembly. Genomics **31:** 301–310.

Olson, M. V., J. E. Dutchik, G. M. Graham, C. Brodeur, M. Helms *et al.*, 1986 Random-clone strategy for genomic restriction mapping in yeast. Proc. Natl. Acad. Sci. USA **83:** 7826–7830.

Perkins, D. D., 2000 *Neurospora crassa* genetic maps and mapped loci. Fungal Genet. Newsl. **47:** 40–58.

Prade, R. A., J. Griffith, K. Kochut, J. Arnold and W. E. Timberlake, 1997 *In vitro* reconstruction of the *Aspergillus nidulans* genome. Proc. Natl. Acad. Sci. USA **94:** 14564–14569.

Sasinowska, H., and M. Sasinowski, 1999 An algorithm for the assembly of robust physical maps based on a combination of multi-level hybridization data and fingerprinting data. Comput. Chem. **23:** 251–262.

Soderlund, C. A., and I. Dunham, 1995 SAM: a system for iteratively building marker maps. Comput. Appl. Biosci. **6:** 645–655.

Soderlund, C. A., and L. P. McGarvan, 1993 *GRAM V1.4: User's Manual.* Los Alamos National Laboratory, Los Alamos, NM.

Soderlund, C. A., O. Longden and R. Mott, 1997 FPC: a system for building contigs from restriction fingerprinted clones. Comput. Appl. Biosci. **13:** 523–535.

Stallings, R. L., D. C. Torney, C. E. Hildebrand, J. L. Longmire, L. L. Deaven *et al.*, 1990 Physical mapping of human chromosomes by repetitive sequence fingerprinting. Proc. Natl. Acad. Sci. USA **87:** 6218–6222.

Sulston, J., F. Mallett, R. Staden, R. Durbin, T. Horsnell *et al.*, 1988 Software for genome mapping by fingerprinting techniques. Comput. Appl. Biosci. **5:** 101–106.

Suyama, A., 1993 ContigMaker: Software tool for contig map construction, pp. 376–384 in *Proc. Genome Informatics Workshop IV*, edited by T. Takagi, H. Imai, S. Miyano, S. Mitaku and M. Kanehisa. Universal Academy Press, Tokyo, Japan.

Tsai, H., and C. Kao, 2000 Using genetic algorithms to construct physical maps of chromosomes with unique probes, pp. 167–168 in *Currents in Computational Molecular Biology*, edited by S. Miyano, R. Shamir and T. Takagi. Universal Academy Press, Tokyo, Japan.

Venter, J. C., H. O. Smith and L. Hood, 1996 A new strategy for genome sequencing. Nature **381:** 364–366.

Wang, Y., R. A. Prade, J. Griffith, W. E. Timberlake and J. Arnold, 1994 A fast random cost algorithm for physical mapping. Proc. Natl. Acad. Sci. USA **91:** 11094–11098.

Communicating editor: J. Arnold