Genomics **66,** 249–256 (2000)
doi:10.1006/geno.2000.6187, available online at http://www.idealibrary.com on IDEAL®

An Algorithm for Clustering cDNA Fingerprints

Erez Hartuv,* Armin O. Schmitt,§[1] Jörg Lange,§[2] Sebastian Meier-Ewert,§[3] Hans Lehrach,§ and Ron Shamir*[4]

*Department of Computer Science, Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978 Israel; and §Max Planck Institute for Molecular Genetics, Ihnestrasse 73, D-14195 Berlin, Germany

Received December 1, 1999; accepted March 8, 2000

Clustering large data sets is a central challenge in gene expression analysis. The hybridization of synthetic oligonucleotides to arrayed cDNAs yields a fingerprint for each cDNA clone. Cluster analysis of these fingerprints can identify clones corresponding to the same gene. We have developed a novel algorithm for cluster analysis that is based on graph theoretic techniques. Unlike other methods, it does not assume that the clusters are hierarchically structured and does not require prior knowledge on the number of clusters. In tests with simulated libraries the algorithm outperformed the Greedy method and demonstrated high speed and robustness to high error rate. Good solution quality was also obtained in a blind test on real cDNA fingerprints. © 2000 Academic Press

INTRODUCTION

Information on the expression levels of genes under various conditions is key to elucidating their function. One way to measure gene expression levels is by sampling cDNAs from the tissue and measuring the amount of cDNA of each gene in the sample. If the cDNAs are picked at random, the abundance of cDNAs extracted indicates the relative expression levels of their genes.

Out of about 100,000 different human genes, the number of genes active in a human cell at any time is over 10,000 (Kinzler *et al.*, 1997). The relative abundance of cDNAs of different genes may vary by a factor of 10,000. This clearly indicates that the size of the sample of cDNAs that must be extracted from a cell to obtain adequate representation of low-abundance genes must be on the order of 100,000 or more.

Sequencing some 100,000 cDNAs per sample is slow and prohibitively expensive. It is also very inefficient, as the high-abundance transcripts are resequenced again and again. Normalized libraries do not fully overcome this redundancy (Bonaldo *et al.*, 1996). Oligo-fingerprinting was developed as an alternative approach (Lennon and Lehrach, 1991; Crkvenjakov *et al.*, 1991; Vicentic *et al.*, 1992; Drmanac and Drmanac, 1994; Drmanac *et al.*, 1996; Meier-Ewert *et al.*, 1994; Milosavljevic *et al.*, 1995). It is based on spotting poly(dT)-primed cDNAs on high-density filters. When a short synthetic oligonucleotide probe hybridizes with the filter under stringent conditions, one obtains a positive hybridization signal with all cDNA clones that contain a DNA sequence complementary to that of the probe. By repeating the experiment with different probes, one obtains for each clone a fingerprint vector, indicating its hybridization level with each probe. cDNAs originating from the same gene have similar oligomer contents and thus should have similar fingerprints. Based on the fingerprints, one can devise computer algorithms to identify and cluster cDNAs originating from the same gene. As a result, ideally, only one cDNA will have to be sequenced from each cluster, and the cluster size will tell the abundance of its gene. Good algorithms must overcome practical difficulties, which include error-prone fingerprints and substantial variability in cDNA length (Meier-Ewert *et al.*, 1998).

Alternative technologies such as DNA microchips (Fodor *et al.*, 1993; Schena *et al.*, 1996) have the advantage of being able to determine the expression levels of thousands of genes in parallel through a single hybridization. However, they are not applicable in all cases, as their application requires that the gene/ORF ensemble be known exactly in advance, and much larger amounts of tissue RNA are required for a single analysis. While the sensitivity is increasingly improved, the oligo-fingerprinting approach is currently one of the most effective strategies for the analysis of novel genes or organisms with few identified genes. Large sequencing projects have successfully identified the majority of human genes and will undoubtedly do the same for a limited number of model organisms. However, it is highly unlikely that the same resources

[1] Present address: metaGen Gesellschaft für Genomforschung, Ihnestrasse 63, 14195 Berlin, Germany.

[2] Present address: Functional Genomics/Life Science Informatics, Novartis Pharma AG, Lichtstrasse, CH-4002 Basel, Switzerland.

[3] Present address: GPC AG, Fraunhofer Strasse 20, D-82152 Martinsried, Germany.

[4] To whom correspondence should be addressed. Fax: 972-3-6409357. E-mail: shamir@math.tau.ac.il.

0888-7543/00 $35.00

will be made available in the near future for sequencing other important model genomes such as *Amphioxus,* sea urchin, and a number of plants and fungi. For such species, oligofingerprinting is an effective means to large-scale gene discovery.

We present here a new algorithm for clustering, called HCS (abbreviation of highly connected subgraphs). The algorithm has been tested intensively on simulated data and was shown to give good results even in the presence of relatively high levels of noise. It was also shown to outperform a central algorithm that has been used for expression analysis (Milosavljevic *et al.*, 1995). In a blind test on a real data set consisting of 2329 cDNAs, the algorithm achieved very good results.

Several graph theoretic approaches to cluster analysis have been suggested (see, e.g., Mirkin, 1996; Matula, 1972; Hansen and Jaumard, 1997). Those include finding connected components, strongly connected components in directed graphs, cliques, and maximal cliques. For a critique of these approaches, see Matula (1970). Our approach is different and is based on repeated minimum-cut computations. Two other approaches that are more similar to ours were proposed by Matula (1969, 1970, 1972, 1977) and by Wu and Leahy (1993). Both of these approaches lack our important stopping criterion, with the ensuing provable results on the clustering quality. In particular, these algorithms do not guarantee a key property of HCS solutions: clusters have diameter 2; i.e., two elements in the same cluster have a high degree of similarity to each other or to a common third member of that cluster. Wu and Leahy's algorithm requires that the number of clusters be known in advance, and all the cuts are computed with respect to the same original graph. For other limitations of these methods, see Hartuv (1998).

For the specific problem of clustering cDNA fingerprints, several approaches were suggested previously: Drmanac *et al.* (1996) build clusters around connected components in the similarity graph. In that graph, vertices correspond to cDNAs and edges correspond to pairs whose similarity is above a threshold θ (see Materials and Methods for definitions). Even with a low false-positive rate in the data, such an algorithm would incorrectly connect true clusters. The way to avoid this is by increasing sharply the threshold for similarity, which causes the splitting of many clusters. Meier-Ewert *et al.* (1998) and Poustka *et al.* (1999) build clusters by computing all maximal cliques and merging maximal cliques with sufficiently large overlap into a single cluster. Computing all maximal cliques is computationally intractable (Garey and Johnson, 1979). Moreover, a high false-negative rate may break large clusters into many maximal cliques with a complicated and hard-to-detect overlap structure. Milosavljevic *et al.* (1995) build clusters using the Greedy algorithm. In each step a new seed clone is chosen, and all clones that are sufficiently similar to the seed are added to its cluster and removed from the data set. To merge falsely separated clusters, the algorithm is run twice: Before the second phase, an average fingerprint is computed for each cluster, and the fingerprints of all the clones in that cluster are replaced by it. Like most Greedy approaches, this algorithm cannot handle well high noise levels, and the quality of its results is very sensitive to the starting point. Meier-Ewert *et al.* (1998) use an algorithm that combines ideas from the two previous methods. Our approach is completely different from all previous cDNA clustering methods. We shall show real data and simulation results that demonstrate the superiority of our algorithm over the Greedy algorithm.

MATERIALS AND METHODS

Library and Fingerprint Construction

The 2329 cDNAs originating from 18 genes used in testing the clustering algorithm were part of a library of some 100,000 cDNAs prepared from purified peripheral blood monocytes. These were isolated, and the cDNA was synthesized by Will Phares (Novartis Forschungsinstitut, Vienna, Austria). cDNAs were synthesized by oligo(dT) priming, cloned into the plasmid vector pSPORT1 (Life Technologies) and transformed into DH10B *Escherichia coli* cells (Life Technologies). Individual colonies were plated out on agar plates and picked into microtiter plates using a Q-bot robot (Genetix, Dorset, UK). A total of 100,000 primary cDNA clones were picked and then arrayed at high density onto nylon membranes using the same Q-bot device.

High-density arrays were then hybridized with whole cDNA clones (also oligo(dT) primed, average length ~1000 nt) whose identity had previously been determined by sequencing (see Table 1). Probes were labeled by random hexamer priming using [α-^{33}P]dATP. Hybridizations were carried out in 0.5 M sodium phosphate, pH 7.2, 7% SDS, at 65°C for at least 3 h and then washed twice at high stringency in 40 mM sodium phosphate, pH 7.2, 0.1% SDS, at 65°C, for 30 min. Positive hybridization signals were scored on three intensity levels, and only those with the highest scores were considered for the purpose of this analysis. By this approach we were able to identify 2329 clones representing the 18 selected gene sequences in the entire library of 100,000 cDNAs. Hybridized membranes were exposed to phosphor storage screens and subsequently scanned using a Phosporimager (Molecular Dynamics, Sunnyvale, CA). Resulting hybridization image files were then analyzed, and positive clones were identified using custom-written software (VisualGrid, available as free download from http://www.gpc-ag.com).

Oligonucleotide fingerprints were generated as described in Meier-Ewert *et al.* (1998), by successive hybridization of 139 decamer oligonucleotides. Each decamer oligonucleotide probe was in fact a pool of 16 decamers that share a common 8 nt core sequence (i.e. NxxxxxxxxN, where N is any nucleotide, and xxxxxxxx is the specific core sequence). The fingerprints of all the cDNAs that were positive in the back hybridizations with the 18 gene-specific probes were then input into the clustering algorithm.

Hybridization Data Preprocessing

Hybridization intensities were renormalized as described in Meier-Ewert *et al.* (1998). From the fingerprints, a real-valued matrix S was formed, containing similarity values between all pairs of cDNAs, with values ranging from 3.42 to 139. The graph G_θ used by the algorithm had a vertex for each clone and an edge connecting two clones if and only if their similarity exceeded $\theta = 110$ (see details below).

Simulation Set-Up

The simulation process receives as input the following parameters: The number of genes in the experiment is N_{genes}. Gene i has C_i copies, so C_i is the true size of the cluster of gene i. Hence, the total number of clones in the simulation is $n = \sum_{i=1}^{N_{genes}} C_i$. L_a and L_b are the minimum and maximum possible lengths, respectively, of a cDNA. Clone lengths are generated according to a normal distribution with mean $\mu = (L_a + L_b)/2$ and standard deviation $\sigma = (L_b + L_a)/6$. The number of probes is p. Probes are assumed to occur along a gene with Poisson distribution with rate λ. This assumption was originally suggested in Michiels *et al.* (1987) and was adopted by other researchers (Alizadeh *et al.*, 1995; Platt and Dix, 1997; Mayraz and Shamir, 1999). The probability that an oligonucleotide occurrence did not register (false-negative hybridization probability) is α. False-positive hybridizations are assumed to have Poisson distribution with rate β. All probe occurrences and error events are assumed to be independent. The result of the simulation is an $n \times p$ hybridization matrix H, in which $H_{ij} = 1$ if clone i hybridized with probe j and $H_{ij} = -1$ otherwise.

We note that the probabilistic assumptions are used only to generate the data for the simulations and are not used by the HCS algorithm. The only assumption that the algorithm makes is that fingerprints of clones from the same cluster tend to have higher similarity values.

The HCS Clustering Algorithm

The input to the clustering process is the $n \times p$ hybridization matrix H, in which rows correspond to the cDNA clones in the library, and columns correspond to the probes. $H_{i,j}$ is the intensity level of the hybridization of clone i with probe j. The ith row, H_i, is the fingerprint of the ith clone. Since cDNAs that originate from the same gene have similar fingerprints, a good clustering algorithm should form a partition in which each cluster contains cDNAs originating from the same gene. The matrix H is used to form the $n \times n$ similarity matrix S, where $S_{ij} = \sum_k H_{ik} \cdot H_{jk}$. For a real value θ, the similarity graph G_θ is a graph with vertices corresponding to the clones and edges connecting clones whose pairwise similarity is at least θ (Matula, 1977; Mirkin, 1996).

We provide some standard graph theoretic terminology needed to describe the algorithm: A cut in a graph is a set of edges whose removal disconnects the graph. The connectivity of a graph G, denoted $k(G)$, is the minimum size of a cut in G. A minimum cut is a cut with minimum number of edges. Several algorithms are available for efficient computation of a minimum cut (Ahuja *et al.*, 1993). Our algorithm is based on the following key observations: A group of clones originating from the same gene should form a subgraph with a relatively high connectivity value. In contrast, the subgraph formed by clones from different genes should have lower connectivity.

The clustering algorithm works on the similarity graph. Had the similarity graph represented the cluster structure perfectly, each cluster would have been a clique, as all members of a cluster are highly similar, and no two clusters would be connected by an edge, as elements from distinct clusters are supposed to be dissimilar. In reality, searching for cliques in the graph would fail in two ways: First, finding maximum cliques is computationally intractable (Garey and Johnson, 1979). Second, and more important, real hybridization matrices contain many errors. Errors in the hybridization data generate inexact fingerprints, leading in turn to errors in the similarity graph: missing edges between vertices in the same cluster and false (extra) edges between vertices in different clusters. Our algorithm was designed to withstand high error rate and work in low-degree polynomial time.

Below we give a high-level description of the algorithm. The detailed exposition and proofs of the algorithm's mathematical properties are given elsewhere (Hartuv and Shamir, submitted for publication). A key definition for our approach is the following: A graph G with $n > 1$ vertices is called highly connected if $k(G) > n/2$. A highly connected subgraph is an induced subgraph that is highly connected. Our algorithm identifies highly connected subgraphs as clusters. The basic HCS algorithm is recursive: On an input graph G, it determines whether that graph is highly connected by computing a minimum cut in G. If G is highly connected, then its vertices form a cluster and the algorithm halts. Otherwise, the edges of a minimum cut are removed, forming two connected components, and the algorithm continues recursively on each component.

The running time of the basic HCS algorithm is bounded by $2N \times f(n,m)$, where $f(n,m)$ is the time complexity of computing a minimum cut in a graph with n vertices and m edges, and N is the number of clusters. Typically $N \ll n$. The best deterministic time bound known for $f(n,m)$ is $O(nm)$ (Matula, 1987; Nagamochi and Ibaraki, 1992).

Improvements

Singleton adoption. The basic HCS algorithm may leave certain vertices as unclustered singletons. Subsequently, each singleton is checked whether it fits into one of the clusters. For each singleton x, we compute the number of neighbors it has in each cluster and in the singleton set S. If the maximum number of neighbors is sufficiently large, and is obtained by a cluster (and not by S), then x is added to that cluster. The process is repeated up to I times ($I = 50$ was used in practice) to accommodate the changes in clusters as a result of previous adoptions.

The low-degree heuristic. When the input graph contains low-degree vertices, initial minimum cut iterations will separate them one by one from the rest of the graph. Removing low-degree vertices from G_θ eliminates such noninformative iterations and significantly reduces the running time. To utilize this idea, the algorithm receives as input a degree sequence, i.e., a decreasing sequence of integers $d_1, d_2, \ldots, d_t$, and performs t major iterations. In major iteration i, we remove all vertices with degrees below d_i and then apply the HCS algorithm, followed by singleton adoption. All clustered vertices are set aside, and the next major iteration is applied to the remaining graph, using the smaller value d_{i+1}.

Cluster merging. To overcome the possibility of cluster splitting, we applied a final cluster-merging step. This step uses the raw fingerprints and was implemented as described in Milosavljevic *et al.* (1995), to facilitate a comparison of the two algorithms. Specifically, an average fingerprint is computed for each cluster, and clusters that have highly similar fingerprints are merged.

Iterated HCS. When there are several cuts attaining the minimum value in the current subgraph, the minimum cut algorithm chooses one arbitrarily. This may cause some splitting of small clusters into singletons. The HCS algorithm can then be reapplied on the subgraph induced by unclustered elements. Iterating this procedure several times overcomes the problem.

Implementation. The simulation algorithm was written in MATLAB. HCS was written in C++ within the LEDA 3.4.1 environment (Mehlhorn and Naher, 1995). The minimum-cut algorithm implemented in LEDA has an $O(nm + n^2 \log n)$ time complexity (Stoer and Wagner, 1997). Average elapsed time on a 194 MHZ SGI challenge L machine with 32 kB instruction cache and 1024 kB main memory was about 43 min for the 2329 clones data set (see Results). Clustering of another 7800 elements in simulation with slightly lower noise levels required only 6 min.

RESULTS

Clustering Real cDNA Data

We tested the algorithm in a blind test on real cDNA data. The input contained hybridization fingerprints of 2329 cDNAs with 139 oligonucleotide probes. The clones originated from 18 different genes. The high-fidelity clustering, obtained by hybridization with long, unique sequences, is given in Table 1. Note the high variability in abundance of genes, ranging from over 700 cDNAs to a single cDNA per gene. The HCS algorithm found 16 clusters and left 206 entities as singletons. The results of the algorithm are shown in Fig. 1 and Table 2. In 13 of the 16 clusters, over 92% of the

TABLE 1

True Clusters and Gene Identities in the Real Data Set

Cluster	Size	Gene name
T_{18}	709	Elongation factor 1 α
T_{17}	285	Clone 190B1
T_{16}	284	Cytochrome Ce oxidase
T_{15}	213	Tubulin β
T_{14}	187	40S Ribosomal protein S6
T_{13}	146	40S Ribosomal protein S3
T_{12}	108	40S Ribosomal protein S4
T_{11}	91	Glyceraldehyde 3-phosphate dehydrogenase (GAPDH)
T_{10}	86	60S Ribosomal protein L4
T_{9}	67	Elongation factor 1 β
T_{8}	43	Calmodulin
T_{7}	39	Heat shock cognate protein KD71
T_{6}	32	Heat shock cognate protein KD90
T_{5}	14	TNF receptor
T_{4}	12	AEBP1
T_{3}	10	Clone 244D14
T_{2}	2	Clone 241F17
T_{1}	1	Anion channel protein

Note. Clusters are numbered by increasing size. For unidentified genes, only their cDNA number is given.

entities belong to the same gene (Table 2). Therefore we call those clusters almost pure. Note the high level of noise (Fig. 1B).

To quantify the quality of a solution in comparison with the true clustering, we used the following measure: Represent a clustering solution of n elements by an $n \times n$ symmetric matrix C, where $C_{ij} = 1$ if i and j belong to the same cluster and $C_{ij} = 0$ otherwise. Given such matrix representations of the true clustering T and any clustering C of the same data set, the Minkowski measure for the quality of C (Sokal, 1977; Jardine and Sibson, 1971) is the normalized distance between the two matrices, $(\|T - C\|)/\|T\|$, where $\|T\| = \sqrt{\sum_i \sum_j T_{i,j}^2}$. Since the matrices are binary, this is simply the number of pairs on which the two solutions disagree (i.e., they are clustered together in one solution but not in the other), normalized according to the true solution. A perfect clustering would thus obtain the score zero.

Table 3 summarizes the quality of clustering results of the algorithm in comparison with the true solution and with two other algorithms. The Minkowski score of the solution formed by the HCS algorithm is 0.71, compared to 0.77 of the Greedy algorithm. (To give maximum power to the Greedy algorithm, we tried several combinations of threshold values for the two phases of the algorithm and chose the combination that minimizes the Minkowski score. For the HCS, we chose a fixed threshold value in a blind manner and optimized the threshold only in the cluster-merging phase; see Materials and Methods.) The application of cluster merging was essential to obtaining a superior score. Although the gap in the Minkowski score is not large, the solutions differ dramatically on other parameters: Greedy generated an excessive number of clusters and left more than twice as many singletons. Note that both algorithms do not assume knowledge of the true number of the clusters.

As another measure to the solution quality, Table 3 also gives the number of false-positive and false-negative errors that correspond to each of the three solutions, as reflected in G_θ. The true (reference) solution had 70% missing edges ("false-negative" errors with respect to the graph G_θ) and 0.7% extra edges ("false positives" in G_θ). In comparison, HCS had 62% missing edges and 0.77% extra edges. Greedy had the lowest missing edges rate but obtained a substantially higher extra edges rate.

As a side remark, we suspect that cluster T_{13}, which was the most fragmented in (40S Ribosomal protein S3) our solution (Table 2), is in fact incorrect. Its inhomogeneity can also be seen in the similarity matrix (Fig. 1B). (See Discussion for further comments.)

Simulation Results

We tested the algorithm on simulated data generated with varying noise levels. The detailed description of the simulation setup is given under Materials and Methods. Figure 2 summarizes the results of systematic experiments with the HCS algorithm on simulated data. Figure 2A shows the performance of the algorithm for various sizes of problems. The results are consistently good, and they improve as the problem size increases, as the effect of the smaller clusters diminishes. Figure 2 also gives results of the Greedy algorithm of Milosavljevic *et al.* (1995) on the same problems. Since only binary fingerprints were generated in the simulation, the cluster-merging phase was not used by both algorithms (see Discussion). The HCS algorithm is consistently superior, and the difference in quality is up to an order of magnitude. Moreover, the solutions of the Greedy algorithm deteriorate as the problem size increases, while those of the HCS algorithm improve. The variance in the quality of the Greedy solutions is substantially larger, due to the extreme dependence of that algorithm on the starting point.

The effective range of the number of probes (Fig. 2B) is 100–300. Increasing the expected false-positive hybridization rate up to 25% has a negligible effect on the quality of the results. In contrast, the false-negative hybridization rate can be increased to 40–50% with little effect. Beyond these values, quality decreases rapidly as the error parameters increase. These results are quite encouraging, as the error rate in real large-scale hybridization experiments is quite high, but it falls within the range giving high-quality clustering results according to our experiments.

DISCUSSION

We have designed a novel algorithm for clustering cDNA fingerprints and have tested it on real and sim-

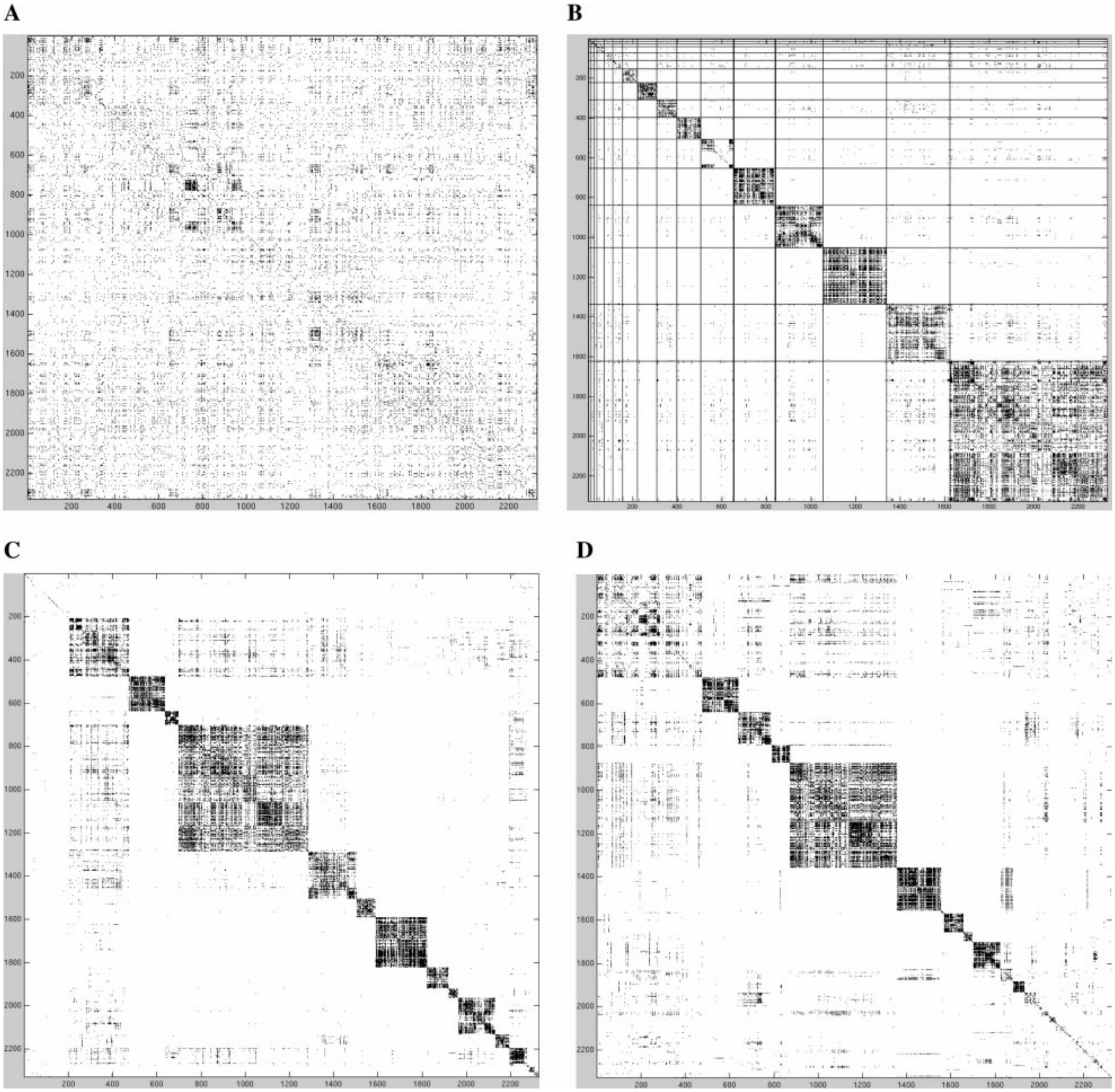

FIG. 1. Clustering results on real cDNA data. (**A**) The binarized similarity matrix S. A black point at position (i,j) indicates that $S_{i,j} \geq 110$. (**B**) Reordering of **A** according to the true clustering. cDNAs from the same true cluster appear consecutively, and black lines delineate borders between different clusters. (**C**) Reordering of **A** according to the clustering produced by the HCS algorithm. cDNAs from the same computed cluster appear consecutively, with no borderlines. Clusters are presented in the order of detection. (**D**) Reordering of **A** according to the solution produced by the Greedy algorithm.

ulated expression data with very good results. Experience with more real data examples would allow even better tuning of the algorithm. The test reported here was the first run of the algorithm on real data for which the solution was known.

Although the solution generated by the algorithm is not perfect, we argue that it is quite good and can serve as a useful tool in gene discovery and expression analysis. By sequencing a small sample from each cluster, one can identify those clusters that are relatively pure and concentrate on sequencing the nonpure and low-abundance clusters only. Another strategy for using the clustering results is to compute the average fingerprint of each cluster and compare it to known gene sequences from databases. This approach was successfully demonstrated by Meier-Ewert *et al.* (1998) and Poustka *et al.* (1999). Using that strategy, most known genes can be detected without any sequencing. This strategy can also detect impure clusters.

Our real data test used as the “true” reference a clustering obtained by hybridization with long (~1000 nt) probes. Though not always completely accurate, that clustering has high fidelity. The stringent hybridization conditions permit the detection of highly com-

TABLE 2

Comparison of the Clusters Formed by the HCS Algorithm with the True Clusters

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	Total
S			7	6	2	9	19	21	9	1	7	5	43	14	9	10	21	23	206
C_1	1	1		6	2	5	13	18	5	2	17	6	16	7	25	9	51	85	269
C_2														**162**					162
C_3													**62**						62
C_4					1	15	1						4			1	1	**563**	587
C_5					5	1	2				4		2		2		**199**	2	217
C_6										**83**			2						85
C_7		1	1	1									2			**224**	2	1	232
C_8												**97**							97
C_9									**42**				1						43
C_{10}															**170**				170
C_{11}											**61**		1						62
C_{12}			1		2	4	4	5		2		7	4	7	4	10	31	81	
C_{13}									**6**										6
C_{14}													1			**26**			27
C_{15}			4										5			4		3	16
C_{16}																**6**			6
Total	1	2	12	14	10	32	39	43	67	86	91	108	146	187	213	284	285	708	2329

Note. $T_1, \ldots, T_{18}$: the true clusters. $C_1, \ldots, C_{16}$: clusters found by the HCS algorithm. *S*: singleton set. Position (i,j) is the number of clones that belong to C_i and T_j. Boldface numbers indicate 92% or more (95% or more except for C_5) of their row totals, indicating pure or almost pure clusters.

plementary sequences only, as false-positive matches require significant homology over stretches of more than 300 nt (Sambrook *et al.*, 1989). Certainly, a perfectly true solution would be to sequence fully all cDNAs used in this analysis, but for quality assessment this reference is quite adequate.

The basic HCS algorithm repeatedly splits the set of clones using minimum cuts until a highly connected subgraph is formed. Such an algorithm has some provable properties that are desirable for clustering (Hartuv and Shamir, submitted for publication). In particular, it guarantees that every cluster has diameter 2, namely, each two elements in the same cluster are highly similar (i.e., with similarity level above θ) or are both highly similar to a common third member of the cluster. In contrast, the union of any two subgraphs split by the algorithm is unlikely to manifest such cohesive properties.

The low-degree heuristic is intended to speed up the basic algorithm. Our experience shows that a judicious choice of the degree sequence has a dramatic effect on the running time with only a minor effect on the clustering quality. For example, on the real data set discussed above, different degree sequences reduce the running time by a factor of 40 and yet lead to extremely similar results (Hartuv, 1998). Selecting the appropriate degree sequence requires some experience or experimentation with the problem data. One useful aid may be the knowledge of the correct clustering on some small subset of the data. Such knowledge is often available for fingerprint data, since some known genes are used as internal controls to monitor the quality of the hybridization process (Meier-Ewert *et al.*, 1998).

A key parameter that influences the clustering quality is the threshold θ. Like the degree sequence, a good value of θ can be determined using a control subset. Our simulations show that the range of θ values that give near-optimal clustering results is quite wide (re-

TABLE 3

Quality of Solutions Given by Three Algorithms on the 2329 Clones Data Set

Solution type	Total edges	Missing edges	Missing edges (%)	Extra edges	Extra edges (%)	No. of clusters	Minkowski	No. of singletons
True	400,278	281,895	70	15,909	0.69	18		
HCS without merge	220,434	114,810	52	28,668	1.15	17	0.83	206
HCS	302,544	186,711	62	18,459	0.77	16	0.71	206
Greedy	182,435	99,523	55	51,380	2.03	66	0.77	478

Note. Total edges: number of intracluster pairs of clones in the solution. Missing edges: intracluster pairs with similarity that falls below the threshold. Extra edges: intercluster pairs that have similarity above the threshold. The threshold for each algorithm was set in a way that optimizes its Minkowski score (the threshold used for the true solution was the same as for the HCS algorithm). True: the true clustering solution. HCS without merge: the HCS algorithm without the cluster merging phase. HCS: the full algorithm. Greedy: the algorithm of Milosavljevic *et al.* (1995). Minkowski: the score of solution quality (see text). Singletons: elements that are left unclustered by the algorithms. Note the extremely high error rate in the true solution.

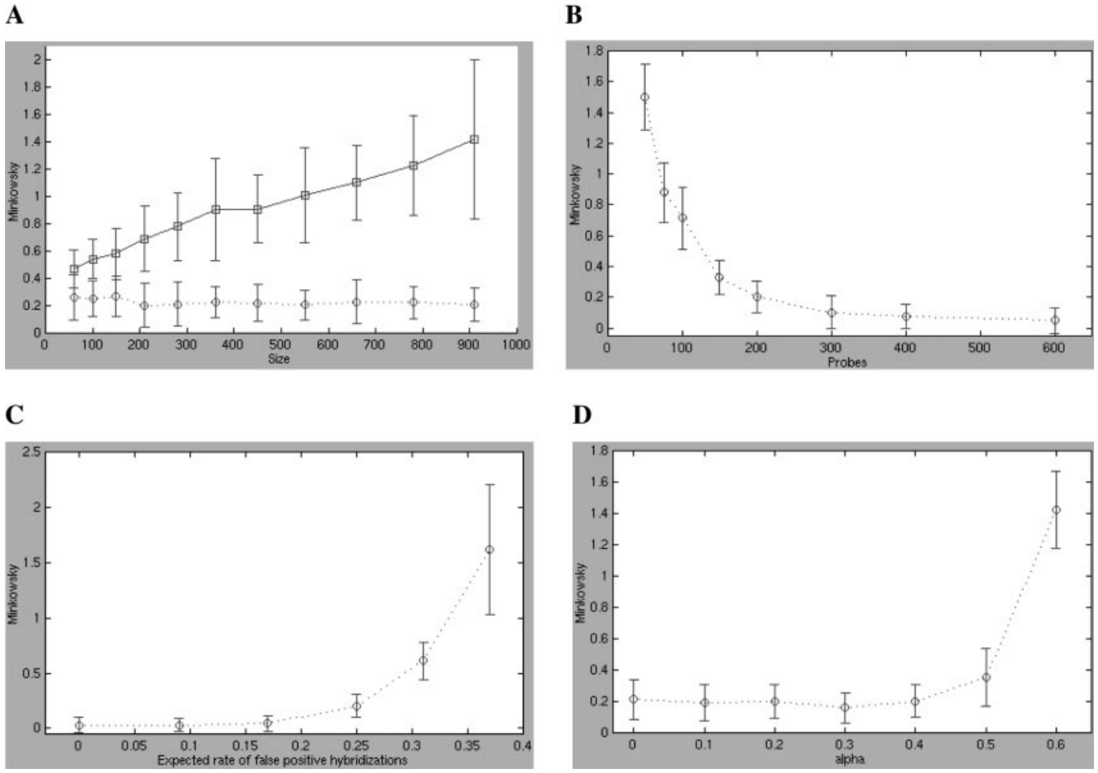

FIG. 2. Impact of problem parameters on the performance of the algorithm on simulated fingerprints, and comparison with the Greedy algorithm. Cluster structure: 450 elements in nine clusters of sizes 10, 20, 30, . . . , 90. The number of probes is 200. $\beta = 0.0015$, so the expected rate of false-positive hybridizations is 25%. $\alpha = 0.4$, so the expected false-negative hybridization rate is 40%. $L_a = 500$ nt; $L_b = 2500$ nt. Poisson rate for probe appearance is $\lambda = 0.005$. In each experiment one parameter value was changed while the rest were kept at the default values. Averages and standard deviations are based on 50 simulations per data point. All results are for the Minkowski score. (**A**) Impact of problem size and comparison to Greedy. Cluster sizes were 10, 20, 30, . . . , 130, and the different problem sizes were obtained by taking the first 3, 4, . . . , 13 sizes. HCS: dotted line; Greedy: continuous line. Error bars denote 1 standard deviation. (**B**) Impact of the number of probes. (**C**) Impact of false-positive rate. (**D**) Impact of false-negative rate.

sults not shown). Consequently, a good value can be rapidly found. The initial value of θ can be based on prior knowledge of the abundance distribution and the expected noise levels.

In our simulation we assumed that all probes have the same rate, which is not satisfied by randomly chosen probes on real gene sequences. However, as was shown by Mayraz and Shamir (1999), this can be remedied by a judicious choice of probes. The results on the real data set also demonstrate that the effect of this problem is not large.

The simulation we have performed demonstrates the robustness of the algorithm to very high noise levels. It is, however, limited to generating only binary fingerprints. This handicaps to some extent the capabilities of the HCS algorithm and to an even larger extent the Greedy algorithm, which depends strongly on the real valued fingerprints. Generating real valued fingerprints would be more realistic, but the complex process of obtaining the real fingerprints is not easy to model in simulation. The results of the algorithm with real valued fingerprints should be at least as good as with binary fingerprints, since more information is exploited.

Additional improvements to the algorithm can be achieved by using a faster minimum cut algorithm (e.g., Karger, 1996) and by attempting to find maximal highly connected subgraphs (e.g., using the cohesiveness function of Matula, 1972). Using a weighted minimum-cut algorithm may also improve the results.

A comparison of our algorithm with classical clustering algorithms like k-means (Hartigan, 1975) may be interesting. One clear advantage of our algorithm is that the number of clusters need not be prespecified. Comparison with classical hierarchical clustering algorithms is also of interest. Finally, although the algorithm we developed was tested in the context of gene expression, its use is not limited to this application, and it can be used to solve other clustering problems.

ACKNOWLEDGMENTS

This research was supported in part by a grant from the German–Israeli Foundation for Scientific Research and Development. R.S. was also supported in part by a grant from the Ministry of Science, Israel. Parts of this work were performed while R.S. was on sabbatical at the Department of Computer Science and Engineering, University of Washington, Seattle. Portions of this paper have been presented at the Third International Conference on Computational Molecular Biology (RECOMB '99) (Hartuv *et al.*, 1999).

REFERENCES

Ahuja, R. K., Magnanti, T. A., and Orlin, J. B. (1993). "Network Flows: Theory, Algorithms and Applications," Prentice Hall, Englewood Cliffs, NJ.

Alizadeh, F., Karp, R. M., Newberg, L. A., and Weisser, D. K. (1995). Physical mapping of chromosomes: A combinatorical problem in molecular biology. *Algorithmica* **13:** 52–76.

Bonaldo, M. F., Lennon, G., and Soares, M. B. (1996). Normalization and subtraction: Two approaches to facilitate gene discovery. *Genome Res.* **6:** 791–806.

Crkvenjakov, R., Drmanac G., Lennon S., Drmanac I., Labat R., and Lehrach, H. (1991). Partial sequencing by oligohybridization: Concept and applications in genome analysis. *In* "Proceedings of the First International Conference on Electrophoresis Supercomputing and the Human Genome" (C. Cantor and H. Lim, Eds.), pp. 60–75, World Scientific, Singapore.

Drmanac, S., and Drmanac, R. (1994). Processing of cDNA and genomic kilobase-size clones for massive screening mapping and sequencing by hybridization. *BioTechniques* **17:** 328–336.

Drmanac, S., Stavropoulos, N. A., Labat, I., Vonau, J., Hauser, B., Soares, M. B., and Drmanac, R. (1996). Gene-representing cDNA clusters defined by hybridization of 57,419 clones from infant brain libraries with short oligonucleotide probes. *Genomics* **37:** 29–40.

Fodor, S. P., Rava, R. P., Huang, X. C., Pease, A. C., Holmes, C. P., and Adams, C. L. (1993). Multiplexed biochemical assays with biological chips. *Nature* **364:** 555–556.

Garey, M. R., and Johnson, D. S. (1979). "Computers and Intractability: A Guide to the Theory of np-Completeness," Freeman, San Francisco.

Hansen, P., and Jaumard, B. (1997). Cluster analysis and mathematical programming. *Math. Programming* **79:** 191–215.

Hartigan, J. A. (1975). "Clustering Algorithms," Wiley, New York.

Hartuv, E. (1998). "Cluster Analysis by Highly Connected Subgraphs with Applications to cDNA Clustering," M.Sc. thesis, Department of Computer Science, Tel Aviv University.

Hartuv, E., and Shamir, R. "A Clustering Algorithm Based on Graph Connectivity," Technical report, Tel Aviv University, Department of Computer Science. Submitted for publication. [Manuscript available at http:www/math.tau.ac.il/∼shamir/papers/hcs.ps]

Hartuv, E., Schmitt, A., Lange, J., Meier-Ewert, S., Lehrach, H., and Shamir, R. (1999). An algorithm for clustering cDNAs for gene expression analysis using short oligonucleotide fingerprints. *In* "Proceedings Third International Symposium on Computational Molecular Biology (RECOMB 99)," pp. 188–197. ACM Press, New York.

Jardine, N., and Sibson, R. (1971). "Mathematical Taxonomy," Wiley, London.

Karger, D. R. (1996). Minimum cuts in near linear time. *In* "Proceedings of the 28th Annual ACM Symposium on Theory of Computing," pp. 56–63.

Kinzler, L., Zhang, W., Zhou, V. E., Velculescu, S. E., Kern, R. H., Hruban, S. R., Hamilton, B., and Vogelstein, K. W. (1997). Gene expression profiles in normal and cancer cells. *Science* **276:** 1268–1272.

Lennon, G. S., and Lehrach, H. (1991). Hybridization analysis of arrayed cDNA libraries. *Trends Genet.* **7:** 60–75.

Matula, D. W. (1969). The cohesive strength of graphs. *In* "The Many Facets of Graph Theory," (G. Chartrand and S. F. Kapoor, Eds.), Lecture Notes in Mathematics No. 110, pp. 215–221, Springer-Verlag, Berlin.

Matula, D. W. (1970). Cluster analysis via graph theoretic techniques. *In* "Proceedings of the Louisiana Conference on Combinatorics, Graph Theory and Computing," (R. C. Mullin, K. B. Reid, and D. P. Roselle, Eds.), pp. 199–212, University of Manitoba, Winnipeg.

Matula, D. W. (1972). k-Components, clusters and slicings in graphs. *Siam J. Appl. Math.* **22**(3): 459–480.

Matula, D. W. (1977). Graph theoretic techniques for cluster analysis algorithms. *In* "Classification and Clustering" (J. Van Ryzin, Ed.), pp. 95–129, Academic Press, San Diego.

Matula, D. W. (1987). Determining edge connectivity in $O(nm)$. *In* "Proceedings of the 28th IEEE Symposium on Foundations of Computer Science," pp. 249–251. Computer Society Press of the IEEE, Washington DC.

Mayraz, G., and Shamir, R. (1999). Construction of physical maps from oligonucleotide fingerprints data. *J. Comput. Biol.* **6**(2): 237–252.

Mehlhorn, and Naher (1995). LEDA: A platform for combinatorial and geometric computing. *Commun. ACM* **38**(1): 96–102.

Meier-Ewert, S., Rothe, J., Mott, R., and Lehrach, H. (1994). Establishing catalogues of expressed sequences by oligonucleotide fingerprinting of cDNA libraries. *In* "Identification of Transcribed Sequences" (U. Hochgeschwender, Ed.), pp. 253–260, Plenum, New York.

Meier-Ewert, S., Lange, J., Gerst, H., Herwig, R., Schmitt, A., Freund, J., Elge, T., Mott, R., Herrmann, B., and Lehrach, H. (1998). Comparative gene expression profiling by oligonucleotide fingerprinting. *Nucleic Acids Res.* **26**(9): 2216–2223.

Michiels, F., Craig, A. G., Zehetner, G., Smith, G. P., and Lehrach, H. (1987). Molecular approaches to genome analysis: A strategy for the construction of ordered overlapping clone libraries. *Cabios* **3**(3): 203–210.

Milosavljevic, A., Strezoska, Z., Zeremski, M., Grujic, D., Paunesku, T., and Crkvenjakov, R. (1995). Clone clustering by hybridization. *Genomics* **27:** 83–89.

Mirkin, B. (1996). "Mathematical Classification and Clustering," Kluwer Academic, Dordrecht/Norwell, MA.

Nagamochi, H., and Ibaraki, T. (1992). Computing edge connectivity in multigraphs and capacitated graphs. *Siam J. Disc. Math.* **5:** 54–66.

Platt, D. M., and Dix, T. I. (1997). Comparison of clone-ordering algorithms used in physical mapping. *Genomics* **40:** 490–492.

Poustka, A. J., Herwig, R., Krause, A., Hennig, S., Meier-Ewert, S., and Lehrach, H. (1999). Toward the gene catalogue of sea urchin development: The construction and analysis of an unfertilized egg cDNA library highly normalized by oligonucleotide fingerprinting. *Genomics* **59:** 122–133.

Sambrook, J., Fritsch, E. F., and Maniatis, T. (1989). "Molecular Cloning: A Laboratory Manual," 2nd ed. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P. O., and Davis, R. W. (1996). Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci. USA* **93:** 10614–10619.

Sokal, R. R. (1977). Clustering and classification: Background and current directions. *In* "Classification and Clustering" (J. Van Ryzin, Ed.), pp. 1–15, Academic Press, San Diego.

Stoer, M., and Wagner, F. (1997). A simple Min-Cut algorithm. *J. ACM* **44**(4): 585–591.

Vicentic, R., Drmanac S., Drmanac I., Labat R., Crkvenjakov A., and Gemmell, A. (1992). Sequencing by hybridization: Towards an automated sequencing of one million M13 clones arrayed on membranes. *Electrophoresis* **13:** 566–573.

Wu, Z., and Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Machine Intelligence* **15**(11): 1101–1113.