2005

MAP2: multiple alignment of syntenic genomic sequences

Liang Ye
Iowa State University

Xiaoqiu Huang
Iowa State University, xqhuang@iastate.edu

MAP2: multiple alignment of syntenic genomic sequences

Abstract
We describe a multiple alignment program named MAP2 based on a generalized pairwise global alignment algorithm for handling long, different intergenic and intragenic regions in genomic sequences. The MAP2 program produces an ordered list of local multiple alignments of similar regions among sequences, where different regions between local alignments are indicated by reporting only similar regions. We propose two similarity measures for the evaluation of the performance of MAP2 and existing multiple alignment programs. Experimental results produced by MAP2 on four real sets of orthologous genomic sequences show that MAP2 rarely missed a block of transitively similar regions and that MAP2 never produced a block of regions that are not transitively similar. Experimental results by MAP2 on six simulated data sets show that MAP2 found the boundaries between similar and different regions precisely. This feature is useful for finding conserved functional elements in genomic sequences. The MAP2 program is freely available in source code form at http://bioinformatics.iastate.edu/aat/sas.html for academic use.

Disciplines
Bioinformatics | Computational Biology | Genetics and Genomics

Comments
This article is published as Ye, Liang, and Xiaoqiu Huang. "MAP2: multiple alignment of syntenic genomic sequences." *Nucleic acids research* 33, no. 1 (2005): 162-170. doi: 10.1093/nar/gki159.

162–170 *Nucleic Acids Research, 2005, Vol. 33, No. 1*
doi:10.1093/nar/gki159

MAP2: multiple alignment of syntenic genomic sequences

Liang Ye and Xiaoqiu Huang*

Department of Computer Science, Iowa State University, 226 Atanasoff Hall, Ames, IA 50011-1040, USA

Received September 11, 2004; Revised November 25, 2004; Accepted December 10, 2004

ABSTRACT

We describe a multiple alignment program named MAP2 based on a generalized pairwise global alignment algorithm for handling long, different intergenic and intragenic regions in genomic sequences. The MAP2 program produces an ordered list of local multiple alignments of similar regions among sequences, where different regions between local alignments are indicated by reporting only similar regions. We propose two similarity measures for the evaluation of the performance of MAP2 and existing multiple alignment programs. Experimental results produced by MAP2 on four real sets of orthologous genomic sequences show that MAP2 rarely missed a block of transitively similar regions and that MAP2 never produced a block of regions that are not transitively similar. Experimental results by MAP2 on six simulated data sets show that MAP2 found the boundaries between similar and different regions precisely. This feature is useful for finding conserved functional elements in genomic sequences. The MAP2 program is freely available in source code form at http://bioinformatics.iastate.edu/aat/sas.html for academic use.

INTRODUCTION

High-quality human genomic sequences have been produced and genomic sequences from other species including mouse, rat, chicken, dog and fish are becoming available. Genomic sequences from other species are useful in understanding of the human genome through comparative analysis (1–8). A fundamental tool in comparative analysis of genomic sequences is a multiple sequence alignment program for comparing genomic sequences. A number of alignment programs have recently been developed to compare genomic sequences of at least 100 kb (9–11). Those recent programs have increased the capacities of the previous programs by two to four orders of magnitude. However, because multiple sequence alignment is a computationally difficult problem (12), continued improvements to the existing techniques are necessary to meet the needs of comparative analysis of genomic sequences.

We describe a multiple alignment algorithm based on a pairwise alignment algorithm recently developed by Huang and Chao (13). The pairwise alignment algorithm extends the Needleman–Wunsch algorithm (14) to handle genomic sequences with similar regions (such as exon and regulatory regions) separated by different regions (such as intron and intergenic regions). In a single step of dynamic programming computation, the pairwise algorithm produces an optimal alignment consisting of an ordered list of aligned similar regions separated by unaligned different regions. The multiple alignment algorithm builds alignments progressively by using the pairwise algorithm, where the pairwise algorithm is guided by intermediate alignments. In the progressive alignment approach, more similar sequences are aligned earlier than less similar sequences (15). When two intermediate alignments are combined into a larger alignment, only the similar regions of the intermediate alignments are candidates for alignment by the pairwise alignment. In other words, the different regions of the intermediate alignments remain as parts of the different regions of the larger alignment. This feature prevents the algorithm from making errors of aligning different regions of one intermediate alignment with similar regions or different regions of the other intermediate alignment.

The algorithm is implemented as a program named MAP2. The MAP2 program requires the conserved blocks to occur in the same order and orientation in all input sequences. Experimental results produced by MAP2 on four real sets of orthologous genomic sequences show that MAP2 rarely missed a block of transitively similar regions and that MAP2 never produced a block of regions that are not transitively similar. Experimental results produced by MAP2 on six simulated data sets show that MAP2 found the boundaries between similar and different regions precisely. This feature is useful for finding conserved functional elements in genomic sequences. Results produced by existing programs are included for comparison.

*To whom correspondence should be addressed. Tel: +1 515 294 2432; Fax: +1 515 294 0258; Email: xqhuang@cs.iastate.edu

METHODS

We define a general multiple sequence alignment and describe an algorithm for computing a general alignment. Then, we present two measures for the evaluation of alignments.

A multiple alignment algorithm

A general alignment of K sequences consists of similarity blocks and difference sections in an increasing order of sequence positions. A similarity block is an ordinary alignment of regions of the sequences, where for each sequence, a non-empty region of the sequence is present in the block. Each region in the similarity block is transitively similar to every other region in the block. A difference section involves some regions of the sequences, where either no region is present in the section for some sequence, or one region in the section is not transitively similar to another region in the section. In other words, similarity blocks are intended to cover regions conserved among all the sequences, whereas difference sections are intended to cover the remaining regions. Only regions in each similarity block are aligned and reported. For genomic DNA sequences, similarity blocks are usually made of conserved exon and regulatory regions, whereas difference sections are usually made of different intron and intergenic regions. An example alignment of four sequences is shown in Figure 1.

A general multiple alignment of K sequences is computed by using the pairwise algorithm of Huang and Chao (13) in the progressive alignment paradigm of Feng and Doolittle (15). The pairwise algorithm computes an optimal alignment of two sequences in linear space and quadratic time, where an alignment consists of similarity blocks separated by difference sections. The pairwise algorithm improves the Needleman–Wunsch algorithm (14) by finding an ordered list of similar regions between the two sequences. This feature is suitable for comparison of genomic sequences with conserved regions separated by different regions.

In the pairwise algorithm of Huang and Chao, a new type of alignment components called difference sections is introduced in addition to the standard components of substitutions and gaps. A difference section consists of one or two sequence regions that are not aligned. A difference section is given a constant penalty. An alignment consists of similarity blocks and difference sections, where a similarity block consists of ordinary substitutions and gaps from the Needleman–Wunsch alignment model. Under the Needleman–Wunsch alignment model, different sequence regions are also aligned, where the penalty of different regions is linear in their lengths. However, under the extended model, long different regions have to be in the difference sections of an optimal alignment. Otherwise, the long different regions would incur heavy penalties and the alignment would not be optimal. In addition, the boundaries of the long different regions are precisely identified and shown on the optimal alignment.

The pairwise algorithm is used in two situations in the progressive alignment paradigm. First, it is used to compute the normalized similarity score (NS score) for each pair of input sequences. The NS score of two sequences is the score of an optimal alignment of the two sequences divided by the length of the alignment. Second, the pairwise algorithm is extended to take as input two alignments and to produce as output an alignment of the two alignments. The extension involves dealing with difference sections in the input alignments and changing substitutions from pairs of bases to pairs of alignment columns. By the definitions of similarity blocks and difference sections, all difference sections in the input alignments must be parts of difference sections in the output alignment, or all similarity blocks in the output alignment must come from parts of similarity blocks in the input alignments. The score of a substitution with two alignment columns is the arithmetic average of all pairwise scores of bases from the columns (16,17).

The algorithm for computing a general multiple alignment of K sequences works in two steps. In Step 1, for each pair of sequences, the NS score of the two sequences in the pair is computed. Then, all the pairs of sequences are arranged in a decreasing order of their NS scores. In Step 2, initially, each sequence is treated as an alignment with just one similarity block. Next, the pairs of sequences are processed one at a time in the above order. For the current pair of sequences, if the two sequences are in different alignments F and G, then an alignment of F and G is constructed, and the alignments F and G are replaced by the resulting alignment. Otherwise, no action is taken. This process terminates when a final alignment of all the

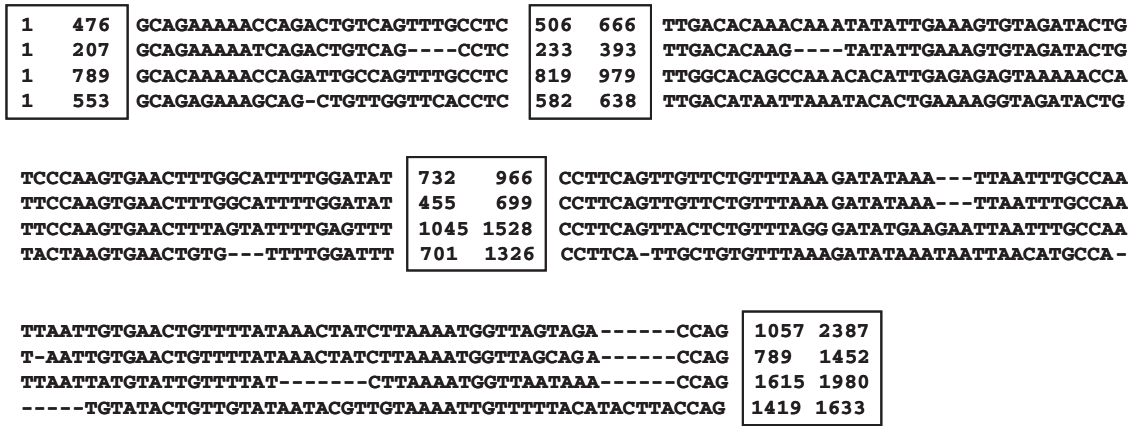

Figure 1. An example alignment of four sequences. Each different section is indicated by a rectangle, which consists of the start and end positions of each region in the difference section.

sequences is constructed. In addition to the final alignment, all the intermediate alignments constructed in Step 2 are saved and reported along with the order in which the alignments are generated. The order is represented by a binary tree, which shows a similarity relationship among the K sequences. An intermediate alignment shows an ordered list of regions conserved among a subset of sequences.

Let N be the total length of the K sequences, and M be the total size of the $K-1$ alignments constructed in Step 2. Then, the time requirement of the algorithm is proportional to N^2 and its space requirement is proportional to $N+M$. Note that each alignment is constructed in linear space. The algorithm is implemented as a C program named MAP2, an improved version of MAP (18).

MAP2 is better at finding boundaries between similar and different regions than existing programs because MAP2 uses the pairwise algorithm of Huang and Chao (13). The existing programs are based on the Needleman–Wunsch algorithm, which produces a global alignment of two entire sequences.

Two similarity measures

An alignment is evaluated by two complementary measures called sum-of-pairs cost and weakest-link percent identity. The sum-of-pairs cost of a column in a similarity block is the number of pairwise differences on the column, where a pairwise difference is a pair of different bases or a pair of a base and a gap symbol. Note that a pair of gap symbols is not counted as a pairwise difference. The average sum-of-pairs cost of an alignment is the sum of sum-of-pairs costs for each column in each similarity block of the alignment divided by the total number of such columns. An alignment with a low average sum-of-pairs cost means that the bases in the alignment are well aligned. The sum-of-pairs measure is a variation of a widely used measure reviewed by Gusfield (19).

A partition of a similarity block involves breaking the regions in the block into two groups. The link percent identity of a partition of a similarity block is the maximum percent identity of induced pairwise alignments linking the two groups. An induced pairwise alignment linking the two groups is obtained from the similarity block by selecting a region from each group and removing all regions except the two selected regions. The weakest-link percent identity of a similarity block is the minimum link percent identity of all partitions of the block. In other words, if a similarity block has a partition of its regions into two groups such that no region in one group is similar to any region in the other group, then the weakest-link percent identity of the block is very low.

We describe a method for computing the weakest-link percent identity of a block. The method is based on finding a maximum-weight spanning tree of a complete graph. In the complete graph, each region of the block is a node and each induced pairwise alignment is an edge with the percent identity of the alignment being the weight of the edge. A spanning tree of the graph is a subgraph that connects all the nodes and has no cycles. The weight of a spanning tree is the sum of weights of every edge in the tree. A maximum-weight spanning tree of the complete graph is a spanning tree of the graph with the maximum weight.

The Kruskal algorithm for finding the edge set of a maximum-weight spanning tree consists of three steps (20). (i) Start with an empty edge set. (ii) Rank the edges of the complete graph in a non-increasing order of weights. (iii) For each edge in the order, add the edge to the edge set unless doing so creates a cycle. The resulting edge set is the edge set of a maximum-weight spanning tree. In Supplementary Material, we show that the weakest-link percent identity of the block is the minimum weight of edges in the maximum-weight spanning tree. An example of the weakest-link percent identity of a similarity block is given in Figure 2.

Note that the progressive multiple alignment method used in the MAP2 program is also related to the Kruskal algorithm for finding a maximum-weight spanning tree of a graph. In this case, each input sequence is a node of the graph and each pairwise alignment is an edge with the score of the alignment being the weight of the edge. The progressive method builds a multiple alignment in a non-increasing order of edge weights.

The two measures are used to evaluate similarity blocks produced by different multiple alignment programs. The sum-of-pairs measure is used in a case where each program produces a block on the same set of regions. A similarity block with the lowest average sum-of-pairs cost is considered to have the highest degree of similarity. However, if some programs produce a similarity block but the other programs do not produce any similarity block on the same set of regions, the sum-of-pairs measure cannot tell whether the regions on a similarity block are transitively similar to each other. The weakest-link measure is used in this case to show whether the regions in a similarity block are transitively similar to each other. Below we explore the relationship between the weakest-link measure and transitive similarity.

For a block of regions, two regions r and s in the block are transitively similar with respect to the block if the induced pairwise alignment of r and s has a percent identity greater than or equal to a percent identity cutoff *pic*, or there is another region m in the block such that r and m are transitively similar with respect to the block and so are m and s. The regions in the block are transitively similar to each other with respect to the block if for every pair of regions in the block, the regions are transitively similar with respect to the block. For example, let $pic = 70\%$ and assume that for a block of three regions a, b and c, the induced pairwise alignment of a and b has a percent identity of 70%, that of b and c has 75% and that of a and c has 50%. Then, the regions in the block are transitively similar to each other with respect to the block.

In Supplementary Material, we show that the weakest-link percent identity of a block is greater than or equal to *pic* if and only if the regions in the block are transitively similar to each other with respect to the block. Thus, if a similarity block has a low weakest-link percent identity, then the transitive similarity relationship among the regions of the block is low and is less likely to be biologically significant. On the other hand, if a similarity block has a high weakest-link percent identity, then the transitive similarity relationship among the regions of the block is high and is more likely to be biologically significant.

RESULTS

The new multiple alignment algorithm described in the last section is implemented as a C program named MAP2. The MAP2 program was evaluated on four real sets of DNA

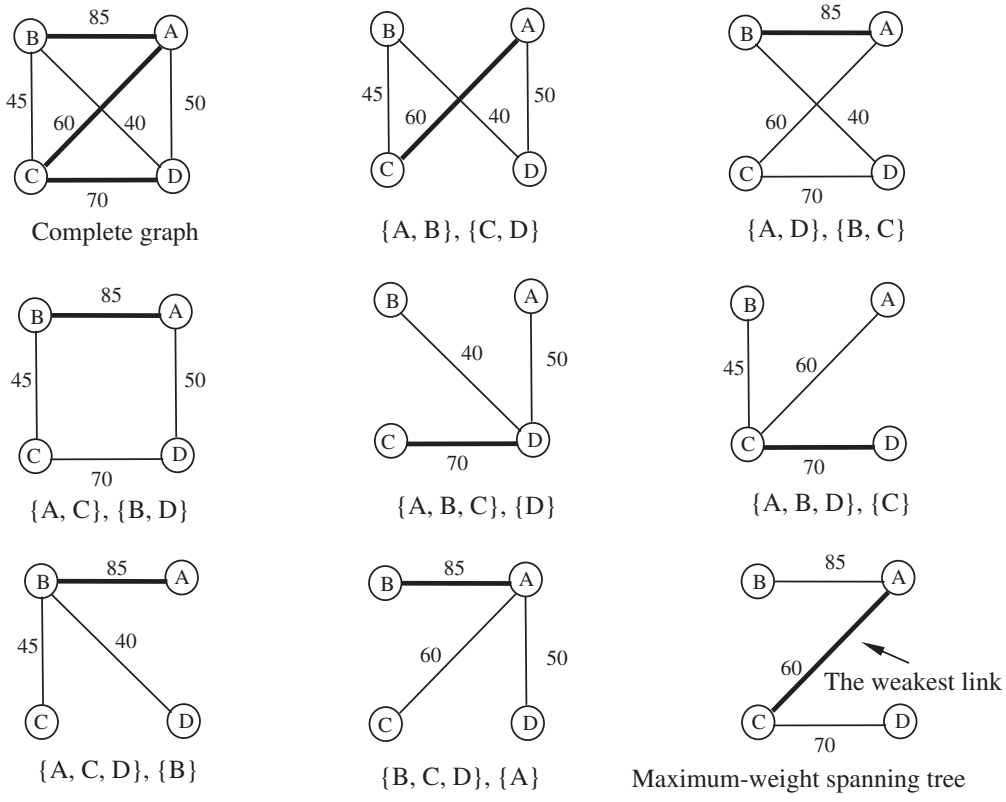

Figure 2. The partitions and the weakest-link percent identity of a similarity block. Each region in the block is represented by a node. For each pair of regions, the induced alignment of the two regions is represented by an edge between the nodes with the percent identity of the alignment shown next to the edge. Each partition link is indicated by a thick edge. The complete graph is on the upper left, a maximum-weight spanning tree of the graph is on the lower right and the rest are every partition and the edges linking the partition.

sequences by the two proposed measures in comparison with existing programs. Set 1 is a group of orthologous genomic sequences harboring the cystic fibrosis *trans*-membrane conductance regulator (CFTR) gene from human, baboon, cow and mouse with an average length of 206 655 bp (21). Set 2 is a group of orthologous genomic sequences harboring the stem cell leukemia (SCL) gene from human, dog, mouse and rat with an average length of 78 544 bp (22). Set 3 is a group of orthologous genomic sequences harboring the met proto-oncogene precursor (MET) gene from human, baboon, cow and mouse with an average length of 516 097 bp (23). Set 4 is a group of orthologous genomic sequences harboring the suppression of tumorigenicity 7 (ST7) gene from human, baboon, cow and mouse with an average length of 536 492 bp (23). The following parameter values were selected for MAP2 based on our experiences with genomic sequence alignment (13,24): match score = 10, mismatch score = −12, difference section score = −250, gap open score = −38 and gap extension score = −3.

Six simulated data sets of four sequences each were constructed and used to evaluate the performance of the four programs in dealing with different regions and finding the boundaries between similar and different regions. The similar regions in each simulated set are complete exon regions of the four sequences in the CFTR set. The different regions between the similar regions in each simulated data set are from genomic regions of Arabidopsis, Drosophila, fish and bacteria/chicken. In other words, each simulated data set of four sequences was constructed by keeping the whole conserved exon regions of the four sequences in the CFTR set and replacing the rest by genomic regions of Arabidopsis, Drosophila, fish and bacteria/chicken. For each group of four similar regions, the percent identity of any two regions in the group is >70%. For each group of four different regions, the percent identity of any two regions in the group is between 34 and 50%.

The following existing multiple alignment programs were included in the comparison: CLUSTALW (25), CHAOS/DIALIGN (10), MLAGAN (9), MAVID (11) and T-COFFEE (26). The CLUSTALW and T-COFFEE programs are designed for protein sequences, whereas the other programs are for genomic DNA sequences. MAVID was run on its web server with the default parameter values. The parameters of MAVID were not made available on the command line by the binary code of MAVID. For the rest of the programs, the programs were run locally. For MLAGAN and CHAOS/DIALIGN, both default and other parameter values were considered. The other parameter values were chosen based on the following rules. If the program has common parameters with MAP2, then comparable values were chosen for the common parameters. For each unique parameter of the program, a value was chosen for the parameter such that based on the documentation on the parameter, the program may be more accurate with the value than with the default value. Both default and the other parameter values were compared on

Table 1. Average sum-of-pairs costs for sets of intersection sub-blocks of multiple alignments from five programs on the CFTR, SCL, MET and ST7 data sets

Program	CFTR				SCL				MET				ST7			
	S50[a]	S65	S67	S70	S33	S36	S37	S39	S223	S258	S283	S315	S260	S295	S320	S353
T-COFFEE[b]	1.574	1.579	1.601	1.610	1.560	1.582	1.579	1.595	1.545	1.553	1.555	1.560	1.509	1.524	1.528	1.537
MAP2	1.595	1.597	1.619	1.627	1.583	1.605	1.602	1.618	1.565	1.574	1.574	1.579	1.531	1.546	1.550	1.559
MLAGAN	1.649	1.648	N/A	N/A	1.604	1.624	1.622	N/A	1.601	1.610	N/A	N/A	1.573	1.586	1.590	N/A
CHAOS/DIALIGN	1.698	1.702	1.722	N/A	1.665	1.687	N/A	N/A	1.686	1.693	1.696	N/A	1.631	1.645	N/A	N/A
MAVID	1.762	N/A	N/A	N/A	1.810	N/A	N/A	N/A	1.719	N/A	N/A	N/A	1.701	N/A	N/A	N/A

[a]The name S50 means that a set of 50 intersection sub-blocks was selected from every multiple alignment. The number 1.649 on row MLAGAN and column S50 is the average sum-of-pairs cost of 50 intersection sub-blocks of blocks of the MLAGAN alignment that have an intersection with blocks of the MAP2 alignment. The mark N/A on row MLAGAN and column S70 means that the MLAGAN alignment does not contain 70 blocks that have an intersection with blocks of the MAP2 alignment.
[b]The set of intersection sub-blocks for T-COFFEE was generated from the corresponding set of blocks for MAP2 by running T-COFFEE, once for each MAP2 block, on the regions of the MAP2 block.

the real data sets, and the parameter values that resulted in better alignments under the two similarity measures were selected for comparison with MAP2. For CHAOS/DIALIGN, the default parameter values were selected, whereas for MLAGAN, the other parameter values were selected.

The selected values for the four common parameters of MLAGAN are match score = 18, mismatch score = −22, gap open score = −68 and gap extension score = −5. Those values were obtained by multiplying the corresponding parameter values of MAP2 by a factor of 1.8. The factor of 1.8 was selected so that the match score for MLAGAN is 18. The selected values for the unique parameters of MLAGAN are lookback distance = 30, maximum gap length = 8, 3-tuple for case 1 = (10, 1, 20), 3-tuple for case 2 = (11, 2, 25), 3-tuple for case 3 = (7, 1, 25) and 3-tuple for case 4 = (6, 1, 25), where the 3-tuple is of the form (wl, nd, sc) with wl standing for word length, nd for number of degeneracy and sc for score cutoff. The different 3-tuples were used by MLAGAN in the four different cases. CHAOS/DIALIGN has 10 unique parameters and has no common parameter with MAP2. The default parameter values for CHAOS/DIALIGN were omitted here.

The CLUSTALW and T-COFFEE programs could not produce an alignment on any of the four real sets of genomic sequences because the programs are designed for protein alignment. Because T-COFFEE is an improvement to CLUSTALW, T-COFFEE was used to serve as a control for the sum-of-pairs measure as follows. The default parameter values for T-COFFEE were used. The regions of an MAP2 block were extracted from the input sequences, an alignment of the regions was produced by T-COFFEE and the average sum-of-pairs cost of the alignment was calculated.

The similarity blocks of a multiple alignment produced by MAP2 on each set of sequences were evaluated by the two measures. Blocks produced by the three existing programs were often much longer than blocks produced by MAP2. A block of an alignment from another program has an intersection with a block of an MAP2 alignment if they share the same region of the human sequence. It may not be possible to require that the blocks share the same set of regions of the input sequences because the bases in the blocks may be aligned differently. The intersection sub-blocks of blocks are parts of the blocks that share the same region of the human sequence. Intersection sub-blocks were evaluated by the sum-of-pairs measure. The remaining parts of the block that have no intersection with any MAP2 block were combined into a block called as an additional block.

Table 2. Total length (bp) of MAP2 blocks and additional blocks from each of three existing programs on the CFTR, SCL, MET and ST7 data sets

Program[a]	CFTR	SCL	MET	ST7
MAP2 blocks	24 825	16 488	159 893	152 023
MLAGAN (a-blocks)	38 583	43 399	147 194	221 645
CHAOS/DIALIGN (a-blocks)	30 337	38 595	134 125	170 338
MAVID (a-blocks)	19 950	27 977	74 756	120 301

[a]The notation 'a-blocks' represents additional blocks, which have no intersection with any MAP2 block.

The three existing programs produced a number of blocks over sequence regions that appear in difference sections of the MAP2 alignment. Those blocks are also called additional blocks. To assess the degree of transitive similarity of the regions in every additional block, the weakest-link percent identity of every additional block from each of the existing programs was calculated. The total length of additional blocks was calculated. The weakest-link percent identity of each MAP2 block was also calculated and so was the total length of MAP2 blocks.

On the CFTR data set, MAP2 produced an alignment with 70 similarity blocks. Of the 70 MAP2 blocks, 67 blocks have an intersection with blocks of the CHAOS/DIALIGN alignment on the CFTR data set. Of the 67 MAP2 blocks, 65 blocks have an intersection with blocks of the MLAGAN alignment. Of the 65 MAP2 blocks, 50 blocks have an intersection with blocks of the MAVID alignment. In other words, 3 MAP2 blocks have no intersection with any CHAOS/DIALIGN blocks, 5 MAP2 blocks have no intersection with any MLAGAN blocks and 20 MAP2 blocks have no intersection with any MAVID blocks. Table 1 shows the average sum-of-pairs costs for the four sets of MAP2 intersection sub-blocks and the corresponding sets of intersection sub-blocks from the three existing programs and T-COFFEE. The results for T-COFFEE were used as an independent control for the sum-of-pairs measure. The sum-of-pairs results by the programs on the SCL, MET and ST7 data sets are also shown in Table 1.

The total length of the 70 MAP2 blocks, the total length of all additional blocks for each of the three existing programs and the results on the SCL, MET and ST7 data sets are reported in Table 2. Figure 3A shows the distribution of the weakest-link percent identities of the 70 MAP2 blocks and the distribution of the weakest-link percent identities of all

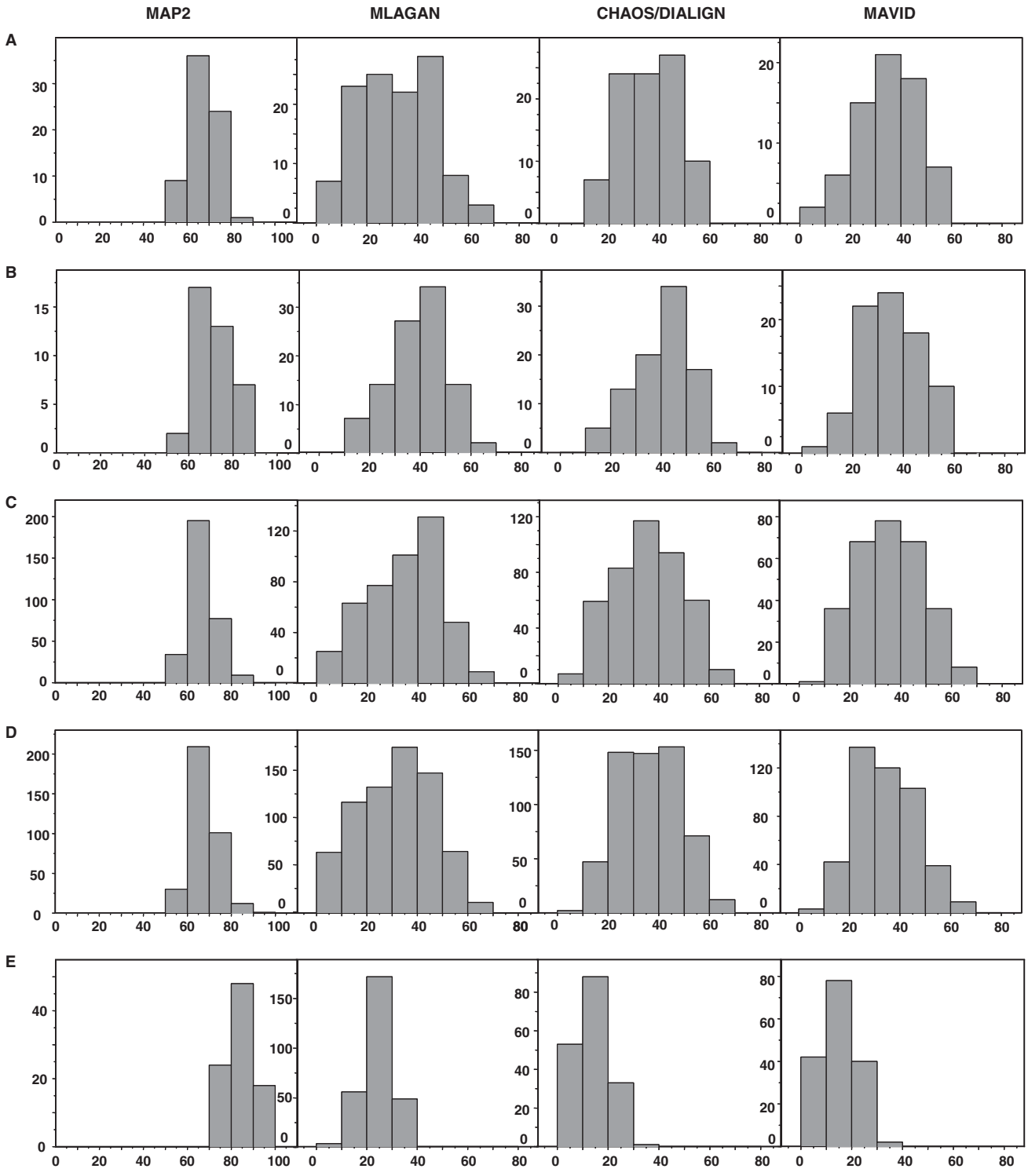

Figure 3. The distribution of the weakest-link percent identities of MAP2 blocks and additional blocks from MLAGAN, CHAOS/DIALIGN and MAVID. A block from the three existing programs is an additional block if the block is not covered by any of the MAP2 blocks. (**A**) The distribution for the CFTR data set. (**B**) The distribution for the SCL data set. (**C**) The distribution for the MET data set. (**D**) The distribution for the ST7 data set. (**E**) The distribution for the simulated data sets.

additional blocks for each of the three existing programs. Figure 3B–D shows results by the programs on the SCL, MET and ST7 data sets. The results indicate that MAP2 rarely missed a block of a weakest-link percent identity >60% and that MAP2 never produced a block of a weakest-link percent identity <50%.

The four programs were run on the six simulated data sets. Figure 3E shows the distribution of the weakest-link percent

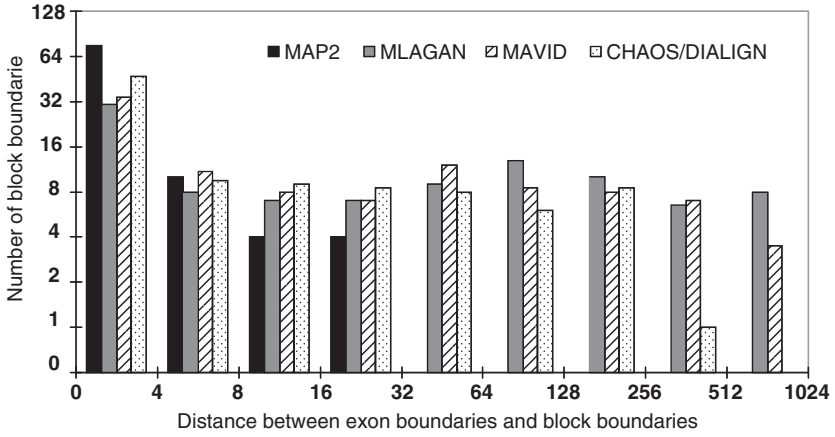

Figure 4. The distribution of the distances between the exon boundaries and block boundaries on the simulated data sets for MAP2, MLAGAN, MAVID and CHAOS/DIALIGN. Exponential scales are used for both directions.

identities of all MAP2 blocks produced on the six simulated data sets and the distribution of the weakest-link percent identities of all additional blocks for each of the other three programs. Each MAP2 block has a weakest-link percent identity of at least 70%. In contrast, there are over 100 additional blocks from each of the other programs and each additional block has a weakest-link percent identity of at most 50%. In other words, MAP2 did not align any group of different regions in the six simulated data sets, whereas each of the other programs aligned many groups of different regions. Each group of different regions is biologically unrelated.

The performance of the programs in finding the boundaries between similar and different regions was evaluated as follows. For every multiple alignment, the distance between each block boundary and its exon boundary was computed. An exon boundary in a multiple alignment of four constructed sequences was defined as the position of the human exon boundary in the constructed sequence. The two boundaries of a block in a multiple alignment of four constructed sequences were defined as the start and end positions of a region of the block in the constructed human sequence. Figure 4 shows the distribution of the distances between block boundaries and their exon boundaries for each of the four programs. For each program, a block with the longest distance from one of its boundaries to an exon boundary was selected and shown in Supplementary Figure 1. The longest distances are 21 bp for MAP2, 768 bp for MLAGAN, 770 bp for MAVID and 297 bp for CHAOS/DIALIGN. For MAP2, the block boundary is 21 bp inside the exon regions as the exon regions are less conserved at the boundary. For each of the other programs, the block boundary is inside the different regions.

The actual or estimated running times of the programs on the four real sets are shown in Table 3. MAP2, CHAOS/DIALIGN and MLAGAN were run locally on a Dell Linux computer with 1 processor of 3.0 GHz and 4 GB of memory, whereas MAVID was run on its web server. It took a few minutes to receive results produced by MAVID. MAP2 was about 3 $\sim$ 5 times slower than CHAOS/DIALIGN.

Table 3. The actual or estimated running times (in minutes) of the programs

Data set	MAP2	MLAGAN	CHAOS/DIALIGN	MAVID
CFTR	298.45	2.68	81.47	$\sim$6
SCL	60.2	0.65	16.5	$\sim$2
MET	2337.07	21.92	526.93	$\sim$12
ST7	2789.78	22.75	648.83	$\sim$12

DISCUSSION

We have developed the MAP2 multiple alignment program based on a generalized pairwise global alignment algorithm for handling long, different intergenic and intragenic regions in genomic sequences. The MAP2 program produces an ordered list of local multiple alignments of similar regions among sequences, where different regions between local alignments are indicated by reporting only similar regions. In addition to the final alignment of all input sequences, MAP2 reports all intermediate alignments that are constructed during the course of generating the final alignment, where each intermediate alignment shows an ordered list of local alignments of similar regions among a subset of input sequences.

We comment on an important feature of MAP2 and its effect on the performance of MAP2. It is shown that for any optimal alignment of two sequences produced by the Huang–Chao algorithm, the score of every similarity block in the alignment is greater than or equal to a non-negative number d, where $-d$ is the score of a difference section (13). The MAP2 alignment algorithm, an extension to the Huang–Chao algorithm, also has this property. For any alignment from MAP2, the score of any similarity block in the alignment is greater than or equal to d. The value used for the parameter d in all MAP2 tests is 250. If each perfect match is given a score of 10, a block of 25 perfect matches has a score of 250.

On a set of sequences, MAP2 produces a major list of ordered blocks among the sequences, with each block having a score greater than or equal to d. If a block of score greater than d is consistent in order with the major list of blocks, then the block is always produced by MAP2. Otherwise, the block

may be missed by MAP2. A block of score less than *d* is never produced by MAP2. A sufficiently long block with a sufficiently high weakest-link percent identity has a score greater than *d*. The block is statistically significant and is likely to be biologically significant because a block of biologically unrelated sequences often has a low weakest-link percent identity. On the other hand, a block of any length with a low weakest-link percent identity has a score less than *d*. The block is not statistically significant and is less likely to be biologically significant.

A sufficiently long block with a medium weakest-link percent identity has a score close to *d*. If the block score is greater than *d* and the block is consistent in order with the major list of blocks, then the block is reported by MAP2. However, if the block score is less than *d*, then the block is missed by MAP2. This is the reason why MAP2 missed some blocks with a weakest-link percent identity around 60% on the test data sets. If a large value is used for the *d* parameter, then MAP2 misses some biologically significant blocks. However, if a small value is used for the *d* parameter, then MAP2 reports a lot of insignificant blocks along with significant blocks. We are currently working on ways to select a proper value for the *d* parameter.

The existing multiple alignment programs are based on the Needleman–Wunsch algorithm. The programs quickly find regions of strong similarity with fast database search methods, use the similar regions as anchors to locate bands of diagonals in the dynamic-programming matrix and compute alignments in the bands with the Needleman–Wunsch algorithm. The pairwise alignment method used in MAP2 searches the entire solution space, whereas the pairwise alignment method in each of the existing programs searches a fraction of the solution space for efficiency. In addition, the method used in MAP2 has a feature to deal with long different regions, whereas the Needleman–Wunsch method lacks the feature. Thus, MAP2 produces much more refined alignments than the existing programs.

It is difficult to evaluate the performance of a multiple genomic sequence alignment program because of lack of large data sets of genomic sequences with experimentally verified annotations of functional elements. However, based on the assumption that sequence similarity may lead to biological significance, every multiple alignment program looks for a similarity relationship among sequences. Thus, we have proposed to evaluate the performance of the MAP2 program by two similarity measures: weakest-link percent identity and sum-of-pairs cost. The weakest-link measure is used to decide whether a group of sequence regions are transitively similar, whereas the sum-of-pairs measure is used to decide whether individual bases of an alignment of transitively similar regions are well aligned.

Experimental results produced by MAP2 on the four real sets of orthologous genomic sequences show that MAP2 rarely missed a block of transitively similar regions and that MAP2 never produced a block of regions that are not transitively similar. A special feature of MAP2 is that MAP2 finds boundaries between similar and different regions precisely. The results indicate that MAP2 meets a selectivity requirement for multiple alignment programs, where similar sequence regions are aligned, but different regions are not aligned (27,28). In contrast, experimental results produced by the existing multiple programs on the four real sets of genomic sequences show that the programs missed a few blocks of transitively similar regions and that the programs produced many blocks of regions that are not transitively similar. Results on the simulated data sets indicate that the existing programs aligned different regions that are not biologically related. In addition, the bases of MAP2 blocks were slightly better aligned than the bases of blocks produced by the existing genomic alignment programs.

A major weakness of MAP2 is its long running time; MAP2 takes 2 days on sequences of 500 kb. Thus, MAP2 is not suitable for alignment on the mega-base level. On the other hand, MAP2 can be used in ordinary laboratories for analysis of genomic sequences of length up to 500 kb. Alignments produced by MAP2 show precisely conserved regions among the sequences, which are usually exon or regulatory regions.

SUPPLEMENTARY MATERIAL

Supplementary Material is available at NAR Online.

ACKNOWLEDGEMENTS

We thank the reviewers for suggestions that significantly improved the presentation of this paper. We also thank Burkhard Morgenstern and Michael Brudno for suggestions on the parameters of CHAOS/DIALIGN and MLAGAN. L.Y. and X.H. were supported in part by NIH Grants R01 HG01502-05 and R01 HG01676-05 from NHGRI. Funding to pay the Open Access publication charges for this article was provided by Iowa State University.

REFERENCES

1. Miller,W. (2001) Comparison of genomic DNA sequences: solved and unsolved problems. *Bioinformatics*, **17**, 391–397.
2. Batzoglou,S., Pachter,L., Mesirov,J.P., Berger,B. and Lander,E.S. (2000) Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res.*, **10**, 950–958.
3. Göttgens,B., Barton,L.M., Gilbert,J.G., Bench,A.J., Sanchez,M.J., Bahn,S., Mistry,S., Grafham,D., McMurray,A., Vaudin,M. *et al.* (2000) Analysis of vertebrate SCL loci identifies conserved enhancers. *Nat. Biotechnol.*, **18**, 181–186.
4. Blanchette,M. and Tompa,M. (2002) Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Res.*, **12**, 739–748.
5. Dermitzakis,E.T., Reymond,A., Lyle,R., Scamuffa,N., Ucla,C., Deutsch,S., Stevenson,B.J., Flegel,V., Bucher,P., Jongeneel,C.V. *et al.* (2002) Numerous potentially functional but non-genic conserved sequences on human chromosome 21. *Nature*, **420**, 578–582.
6. Frazer,K.A., Elnitski,L., Church,D.M., Dubchak,I. and Hardison,R.C. (2003) Cross-species sequence comparisons: a review of methods and available resources. *Genome Res.*, **13**, 1–12.
7. Kellis,M., Patterson,N., Endrizzi,M., Birren,B. and Lander,E.S. (2003) Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, **423**, 241–254.
8. Margulies,E.H., Blanchette,M., Haussler,D. and Green,E.D. (2003) Identification and characterization of multi-species conserved sequences. *Genome Res.*, **13**, 2507–2518.
9. Brudno,M., Do,C.B., Cooper,G.M., Kim,M.F., Davydov,E., Green,E.D., Sidow,A. and Batzoglou,S. (2003) LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res.*, **13**, 721–731.

10. Brudno,M., Chapman,M., Göttgens,B., Batzoglou,S. and Morgenstern,B. (2003) Fast and sensitive multiple alignment of large genomic sequences. *BMC Bioinformatics*, **4**, 66.
11. Bray,N. and Pachter,L. (2004) MAVID: constrained ancestral alignment of multiple sequences. *Genome Res.*, **14**, 693–699.
12. Wang,L. and Jiang,T. (1994) On the complexity of multiple sequence alignment. *J. Comput. Biol.*, **1**, 337–348.
13. Huang,X. and Chao,K.M. (2003) A generalized global alignment algorithm. *Bioinformatics*, **19**, 228–233.
14. Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
15. Feng,D.F. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, **25**, 351–360.
16. Corpet,F. (1988) Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res.*, **16**, 10881–10890.
17. Higgins,D.G. and Sharp,P.M. (1989) Fast and sensitive multiple sequence alignments on a microcomputer. *Comput. Appl. Biosci.*, **5**, 151–153.
18. Huang,X. (1994) On global sequence alignment. *Comput. Appl. Biosci.*, **10**, 227–235.
19. Gusfield,D. (1997) *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, NY.
20. Cormen,T.H., Leiserson,C.E. and Rivest,R.L. (1990) *Introduction to Algorithms*. MIT Press, Cambridge MA.
21. Thomas,J.W. and Touchman,J.W. (2002) Vertebrate genome sequencing: building a backbone for comparative genomics. *Trends Genet.*, **18**, 104–108.
22. Chapman,M.A., Donaldson,I.J., Gilbert,J., Grafham,D., Rogers,J., Green,A.R. and Göttgens,B. (2004) Analysis of multiple genomic sequence alignments: a web resource, online tools, and lessons learned from analysis of mammalian SCL loci. *Genome Res.*, **14**, 313–318.
23. Thomas,J.W., Touchman,J.W., Blakesley,R.W., Bouffard,G.G., Beckstrom-Sternberg,S.M., Margulies,E.H., Blanchette,M., Siepel,A.C., Thomas,P.J., McDowell,J.C. *et al.* (2003) Comparative analyses of multi-species sequences from targeted genomic regions. *Nature*, **424**, 788–793.
24. Huang,X. and Miller,W. (1991) A time-efficient, linear-space local similarity algorithm. *Adv. Appl. Math.*, **12**, 337–357.
25. Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.
26. Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.
27. Briffeuil,P., Baudoux,G., Lambert,C., De Bolle,X., Vinals,C., Feytmans,E. and Depiereux,E. (1998) Comparative analysis of seven multiple protein sequence alignment servers: clues to enhance reliability of predictions. *Bioinformatics*, **14**, 357–366.
28. Morgenstern,B. (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, **15**, 211–218.