BIOINFORMATICS

Vol. 18 no. 6 2002
Pages 777–787

Exon discovery by genomic sequence alignment

Burkhard Morgenstern[1,*], Oliver Rinner[2], Saïd Abdeddaïm[3], Dirk Haase[1], Klaus F. X. Mayer[1], Andreas W. M. Dress[4] and Hans-Werner Mewes[1]

[1]GSF Research Center, MIPS/Institute of Bioinformatics, Ingolstädter Landstraße 1, 85764 Neuherberg, Germany, [2]Physiologisch-Chemisches Institut, Universität Tübingen, Hoppe-Seyler-Straße 4, 72076 Tübingen, Germany, [3]LIFAR-ABISS, Faculté des Sciences et Techniques, Université de Rouen, 76821 Mont-Saint-Aignan Cedex, France and [4]Research Center for Interdisciplinary Studies on Structure Formation (FSPM), Universität Bielefeld, Postfach 100131, 33501, Bielefeld, Germany

Received on July 4, 2001; revised on October 24, 2001; December 10, 2001; accepted on December 20, 2001

ABSTRACT

Motivation: During evolution, functional regions in genomic sequences tend to be more highly conserved than randomly mutating 'junk DNA' so local sequence similarity often indicates biological functionality. This fact can be used to identify functional elements in large eukaryotic DNA sequences by cross-species sequence comparison. In recent years, several gene-prediction methods have been proposed that work by comparing anonymous genomic sequences, for example from human and mouse. The main advantage of these methods is that they are based on simple and generally applicable measures of (local) sequence similarity; unlike standard gene-finding approaches they do not depend on species-specific training data or on the presence of cognate genes in data bases. As all comparative sequence-analysis methods, the new comparative gene-finding approaches critically rely on the quality of the underlying sequence alignments.

Results: Herein, we describe a new implementation of the sequence-alignment program DIALIGN that has been developed for alignment of large genomic sequences. We compare our method to the alignment programs PipMaker, WABA and BLAST and we show that local similarities identified by these programs are highly correlated to protein-coding regions. In our test runs, PipMaker was the most *sensitive* method while DIALIGN was most *specific*.

Availability: The program is downloadable from the DIALIGN home page at http://bibiserv.techfak.uni-bielefeld.de/dialign/

Contact: burkhard@TechFak.Uni-Bielefeld.DE

INTRODUCTION

With the huge amount of genomic data that are now available, *gene prediction* has become a major challenge in computational molecular biology. The goal is to develop computer programs that can automatically identify protein-coding regions in large genomic sequences. Traditionally, there are two distinct approaches to this problem, see, for example, Stormo (2000) or Claverie (1997) for an overview. *Ab initio* or *intrinsic* methods use statistical features such as ORF length, codon frequencies and the location of potential splice sites to distinguish coding from non-coding regions in genome sequences (Krogh *et al.*, 1994; Burge and Karlin, 1997; Lukashin and Borodovsky, 1998). In contrast, *extrinsic* methods try to find similarities between genomic sequences and known proteins (Gish and States, 1993; Gelfand *et al.*, 1996; Birney and Durbin, 2000). Both approaches have advantages and limitations. *Ab initio* methods are able to detect genes with no homologues in protein data bases. It has been shown, however, that the accuracy of these methods is limited (Burge and Karlin, 1998). Also, *ab initio* methods rely on statistical models derived from limited sets of training data, so they tend to be biased towards already known genes (Burset and Guigó, 1996). Homology-based approaches, on the other hand, can reliably identify genes with sufficiently strong similarity to known proteins, but are unable to predict genes to which no homologues are known. Combinations of *ab initio* and *comparative* gene recognition methods have been proposed by Frishman *et al.* (1998) and Usuka and Brendel (2000).

In recent years, a third way of predicting genes and other functional elements in genomic sequences has been emerging: it is possible to identify biologically functional

[*]To whom correspondence should be addressed at Universität Bielefeld, Technische Fakultät, Praktische Informatik, Postfach 100131, 33501 Bielefeld, Germany.

regions by comparing evolutionary related genomic sequences with each other see Wiehe *et al.* (2000) and Miller (2001) for a review. The rationale behind this *phylogenetic footprinting* approach is simple: functional regions are under selective pressure and tend to be more highly conserved than non-functional regions that are subject to random genetic drift, so local sequence similarity usually indicates biological functionality. Several recent studies successfully used cross-species sequence comparison to identify functional regions in large genomic sequences (Ansari-Lari *et al.*, 1998; Jang *et al.*, 1999; Jareborg *et al.*, 1999; Batzoglou *et al.*, 2000; Göttgens *et al.*, 2000; Mallon *et al.*, 2000; Göttgens *et al.*, 2001) and it is now widely accepted that comparative sequence analysis is a powerful and universally applicable tool for genome analysis and annotation. Even the question which genomes should be sequenced next is being discussed in view of the benefits that are to be expected from comparative sequence analysis (Hardison *et al.*, 1997; Miller, 2000).

The phylogenetic footprinting idea can also be applied to the problem of gene identification. A number of novel gene-prediction tools have been developed that are based on comparative analysis of genomic sequences from evolutionary related organisms (Bafna and Huson, 2000; Batzoglou *et al.*, 2000; Korf *et al.*, 2001; Wiehe *et al.*, 2001; Rinner and Morgenstern, 2001; Novichkov *et al.*, 2001). The first and most critical step in sequence comparison is to construct an *alignment* of the sequences in question and the results of any comparative method crucially depend on the discriminative power of the underlying alignments. *Global* pair-wise or multiple alignment procedures such as the Needleman–Wunsch algorithm (Needleman and Wunsch, 1970) or CLUSTAL W (Thompson *et al.*, 1994) are clearly not appropriate for aligning genomic sequences where local homologies are typically separated by large regions of non-related sequences. *Local* methods like BLAST (Altschul *et al.*, 1990), FASTA (Pearson and Lipman, 1988), or Smith–Waterman alignments (Smith and Waterman, 1981) are efficient in detecting isolated *peaks* of local sequence similarity, but they do not give an *overall* picture of the homologies among large sequences.

Therefore, several new alignment programs have been developed that are able to cope with large genomic sequences. The *DNA Block Aligner* (Jareborg *et al.*, 1999) identifies *blocks* of varying degrees of similarity that may contain small gaps; large gaps between these blocks are treated differently to account for long non-conserved regions in the sequences. This tool is specialized for alignment of *non-coding* DNA sequences. Another novel tool for genomic sequence alignment is MUMmer (Delcher *et al.*, 1999). MUMmer is extremely fast and has been successfully used to align entire genomes of closely related species; however, the method seems to have difficulties with aligning sequences from more distantly related organisms, e.g. from primates and rodents.

For gene-prediction purposes, generally applicable alignment methods are needed that can cope with coding as well as non-coding parts of sequences and that are able to align genomic sequences at different evolutionary distances. Such methods are, for example, DIALIGN (Morgenstern *et al.*, 1996; Morgenstern, 1999), PipMaker (Schwartz *et al.*, 2000), GLASS (Batzoglou *et al.*, 2000) and WABA (Kent and Zahler, 2000). DIALIGN constructs pair-wise and multiple alignments from pairs of un-gapped segments of the input sequences, so-called (alignment) *fragments*. PipMaker aligns genomic sequences based on a new implementation of the *Gapped BLAST* algorithm (Altschul *et al.*, 1997; Zhang *et al.*, 1998); local similarities identified by this program are assembled using a *fragment-chaining* algorithm by Zhang *et al.* (1994). GLASS works by recursively aligning matching *k*-mers while WABA breaks large genomic sequences into smaller blocks that are then aligned using hidden Markov models. WABA has been used to align entire eukaryotic genomes consisting of tens of millions of base pairs. Herein, we outline a new version of DIALIGN that has been designed for alignment of large genomic sequences. We use two sets of test data to demonstrate that this method can help to identify exons in eukaryotic genomic sequences of several hundred kb in length purely based on sequence similarity. These results are compared to results that were obtained with PipMaker, WABA and BLAST.

MODIFICATIONS TO THE DIALIGN ALIGNMENT PROGRAM

(a) Sequence similarity at the nucleotide level and at the peptide level

As described in previous papers, DIALIGN constructs alignments as collections of so-called *(alignment) fragments*, i.e. gap-free local pairwise alignments (Morgenstern *et al.*, 1996; Morgenstern, 1999; Abdeddaïm and Morgenstern, 2001). The program assigns a *weight score* to every possible fragment reflecting the degree of similarity among the two segments and then selects a *consistent* set of fragments maximizing their total weight. For *pairwise* alignment, this means that a *chain* of fragments with maximal total weight is selected (Morgenstern, 2000). If DNA sequences are to be aligned, DIALIGN can measure the similarity between two segments in two distinct ways, see Figure 1: the similarity can be assessed at the *nucleotide level* by comparing segments nucleotide-by-nucleotide or at the *peptide level* by translating DNA segments according to the genetic code and then comparing the resulting *peptide segments*

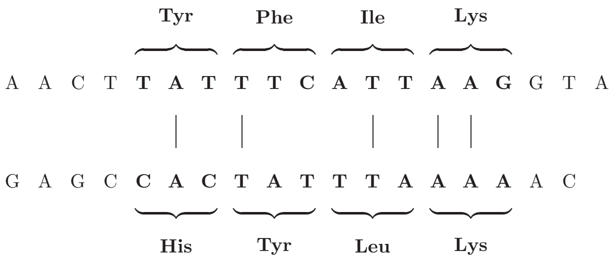
Tyr
Phe
Ile
Lys
A A C T T A T T T C A T T A A G G T A
G A G C C A C T A T T T A A A A A C
His
Tyr
Leu
Lys

Fig. 1. DIALIGN can calculate the *weight score* of a fragment (= gap-free segment pair) at two different levels. At the *nucleotide level*, the number of matching nucleotide pairs is considered and the program calculates the probability of finding a fragment of the respective length with (at least) the same number of nucleotide matches in *random sequences* the same length as the input sequences. The weight of a fragment f is then defined as the negative logarithm of this probability. Alternatively, the score of f can be calculated at the *peptide level*. Here, both segments are translated according to the genetic code, the BLOSUM values of the implied amino-acid pairs are summed up and the program calculates the probability of finding a fragment of the same length with the respective sum of BLOSUM values. The above fragment, for example, would have a low weight score at the *nucleotide level* since it would be rather likely to find a fragment of length 12 with 5 or more matches in random sequences just by chance. Its weight at the *peptide level*, however, would be higher since the four pairs of implied amino acids have high BLOSUM values so one would be less likely to find a segment pair with this sum of BLOSUM values by chance.

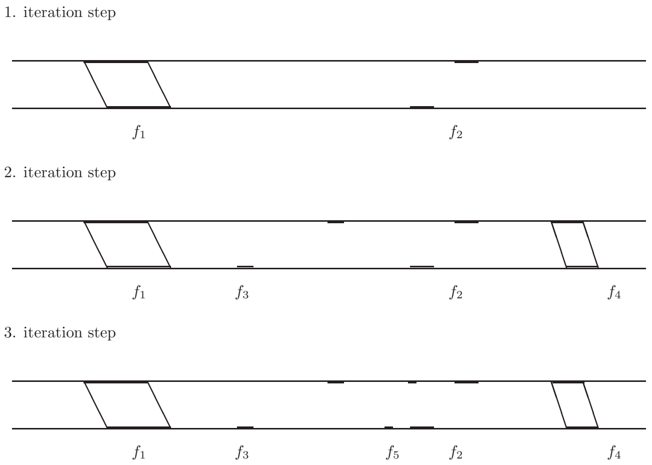

Fig. 2. Iterative scoring scheme for alignment of long genomic sequences. In the first iteration step, the weight score of a *fragment* (gap-free segment pair) is calculated based on the probability of its occurrence in random sequences the same length as the *input sequences*. In subsequent steps, the probability of random occurrence in the intervals between previously detected fragments is considered. In the above example, fragments f_1 and f_2 have been accepted in the first iteration step, based on the probability of random occurrence in sequences the size of the input sequences. In the second iteration step, the weight score of fragment f_3 is calculated based on the probability of finding such a fragment in the space that is left between fragments f_1 and f_2. This way, homologies that do not appear statistically significant if the input sequences are considered as a whole can be detected if the *reduced* space between already accepted fragments is considered.

using the BLOSUM 62 substitution matrix (Henikoff and Henikoff, 1992). In both cases, the program calculates the probability of fragments of the same length and (at least) the same sum of matches or BLOSUM scores, respectively, to occur by chance in random sequences. The weight scores are based on these probabilities, see Morgenstern (1999) for more details.

With previous versions of DIALIGN, the user had to decide whether the similarity of DNA segments was to be assessed at the nucleotide level or at the peptide level. This is not adequate where large genomic sequences are aligned since *coding regions* tend to be more highly conserved at the *peptide level* whereas *non-coding* functional elements are conserved at the *nucleotide level*. Consequently, in the new version of the program, segment pairs can be compared *simultaneously* at both levels. In addition, the peptide-level similarity is calculated for both possible orientations, i.e. for the *plus strand* and for the *reverse complement*. For every fragment (segment pair) f, all three respective similarity values are calculated, and the score of f is then defined to be the maximum of these three values. As a result, the program can now produce *mixed alignments* that consist of *nucleotide fragments* and *peptide fragments* in both orientations depending on which respective type of local similarity is stronger. In the the context of this *mixed alignment* option, we use the terms *N-fragments* for segment pairs with stronger similarity at the *nucleotide level* and *P-fragments* for segment pairs with stronger similarity at the *peptide level*.

(b) Iterative alignment procedure

Previously, DIALIGN computed the similarity score of a fragment in terms of the probability of its random occurrence in sequences *the size of the input sequences* and positive scores were assigned only to those fragments that are rather unlikely to occur by chance. While this approach works well with sequences of moderate size it is not sensitive enough to detect small functional elements in large genomic sequences. In the new release of the program, a more sophisticated iterative procedure can be applied: in a *first* step, the fragments are scored as explained above and a chain of fragments with sufficiently high scores is selected based on this scoring scheme. In *subsequent* steps, intervals between those fragments that have been selected in previous steps are *realigned*. Here, fragments are assessed based on the probability of occurrence in the respective *intervals*, i.e. in the space that is left between the segment pairs that have been previously aligned. This way, segment pairs that do not appear to be statistically significant if the input sequences as a whole are considered can be detected if other parts of the sequences have already been aligned. The number of iteration steps can be specified by the user; to limit accumulation of spurious random fragments, the current implementation of the algorithm uses a default value of three iteration steps.

(c) Anchored alignments

DIALIGN has originally been developed to study protein and DNA sequences of relatively small size (e.g. Morgenstern and Atchley, 2001). During the pair-wise alignment procedure, the program compares each segment within the first sequence—up to a certain maximum segment length—to each segment of the same length within the second sequence. Consequently, the program running time for aligning two sequences was proportional to the product of the sequence lengths (times the maximum segment length) which makes it difficult to apply the method to sequences of more than a few hundred *kb* in length. One possible way of improving the time efficiency of the program is to *anchor* the alignment at regions of strong sequence similarity thereby reducing the search space for the alignment procedure.

If anchoring positions $x_0, \dots, x_N$ in sequence 1 and $y_0, \dots, y_N$ in sequence 2 are used such that the program is forced to align position x_i to position y_i, $0 \leq i \leq N$, the search space for the alignment procedure is reduced to $\sum_{i=1}^{N}(x_i - x_{i-1}) \times (y_i - y_{i-1})$ compared to the product of the sequence lengths for the non-anchored procedure. With the new implementation of DIALIGN, it is possible to anchor alignments at arbitrary user-defined points. Such anchoring points can be found, for example, by fast sequence-comparison methods such as BLAST (Altschul *et al.*, 1990), REPuter (Kurtz and Schleiermacher, 1999; Kurtz *et al.*, 2000) or CHAOS (Brudno and Morgenstern, 2002). Preliminary results indicate that this way the program running time can be reduced by around 95% while the quality of the resulting alignments drops only by 1–2% (M.Brudno and B.Morgenstern, unpublished results). The anchoring option can also be used to include expert knowledge about known homologies in order to improve the biological quality of the resulting alignments.

TEST RESULTS

To test our method, we used benchmark data from two different sources. First, we aligned 42 pairs of genomic sequences from human and mouse compiled by Jareborg *et al.* (1999). These sequences vary in length between less than 6 kb and more than 227 kb (average length 38 kb); they contain a total of 77 known gene pairs. As a second example, we aligned a recently published 105 *kb* segment from tomato (Ku *et al.*, 2000) to a syntenic 32 *kb* segment from *Arabidopsis thaliana* (The *Arabidopsis Thaliana Initiative*, 2000) that we identified by BLAST searches. Here, the *A. thaliana* sequence contains nine known genes with a total of 44 exons but not all of these genes have homologues in the tomato sequence. After removeing low-complexity regions with the RepeatMasker software (Smit and Green, RepeatMasker at http://repeatmasker.genome.washington.edu/cgi-bin/RepeatMasker), we aligned these data sets with DIALIGN, PipMaker, WABA, BLASTN and TBLASTX. A common feature of these programs is that they clearly distinguish between conserved and non-conserved regions in the output alignments so we could evaluate them by comparing conserved regions that they detected in our test sequences to known exons. For the human–mouse example, we used the human sequence as reference, for the tomato–*Arabidopsis* example, the reference was *Arabidopsis*.

The programs in our study are general-purpose alignment programs that are based on universally applicable measures of sequence similarity so they cannot be expected to precisely delimit the boundaries of protein-coding regions. Therefore, we evaluated these programs at the *nucleotide level*, i.e. we considered nucleotides that are part of identified sequence similarities as *true positives* (TP) if they also belong to annotated exons and as *false positives* (FP) if they do not; true and false negatives (TN and FN) are defined accordingly. We used standard measures for prediction accuracy, namely *sensitivity* $Sn = TP/(TP + FN)$, *specificity* $Sp = TP/(TP + FP)$, and *approximate correlation* $AC = 0.5((TP/(TP + FN) + (TP/(TP + FP) + (TN/(TN + FP) + (TN/(TN + FN)) - 1$.

With the new features of DIALIGN described in the previous section, the program distinguishes between two levels of sequence similarity, namely similarity at the *nucleotide level* (N-fragments) and similarity at the *peptide level* (P-fragments). In addition, one can distinguish between fragments returned in the first iteration steps (stronger sequence similarity) and those identified in subsequent steps (weaker similarity). To study how these different types of fragments are correlated with protein-coding regions, we evaluated our results at different levels. First, we evaluated the program by considering *all* fragments returned by the program—i.e. P-fragments and N-fragments—then we ignored N-fragments and considered P-fragments only. To study the effect of the iterative procedure, we did these experiments (a) by considering fragments returned in all *three* iteration steps that the program performs by default and (b) by considering only those fragments that were returned in the *first* iteration step.

WABA also distinguishes between different levels of sequence similarity. The hidden Markov model used by this program has three different *states* to model sequence homology, namely *coding* regions (C), *high similarity* (H) and *low similarity* (L). As with DIALIGN, we looked at these different types of similarities separately and we evaluated WABA in two different ways, namely first by considering only those regions that are characterized as *coding* (C) and then by considering regions characterized as either *coding* or *high similarity* (C + H).

DIALIGN, PipMaker and BLAST produce lists of conserved segment pairs (fragments) each of which is associated with some similarity score. Therefore it is straight forward to filter out weaker similarities by applying cut-off values and to evaluate only those fragments that have similarity scores above these values. For the fragments identified by DIALIGN, we used the *weight scores* described in the *methods* section as a filtering criterion; we used threshold values between 0 and 30 and ignored all fragments with weight scores below these values. For PipMaker, we considered the *percentage-of-identity (PI)* values that the program associates with the returned fragments. PipMaker uses these values to graphically represent output alignments as *Percent-of-Identity Plots* or *PIPs*. The *PI* scores, however, turned out to be unsuitable as cut-off criteria—the reason for this is that this scoring scheme does not take into account the length of a fragment so, for example, a short fragment with 80% identity would have the same score as a long fragment with 80% identity. Therefore, instead of using the *PI* values directly, we used fragment scores *sc* defined by

$$sc = \#\text{matches} - \#\text{mismatches} = len \times (2 \times PI - 100)/100$$

were #matches and #mismatches denote the sum of matches and mismatches, respectively, in a fragment and *len* is its length. For PipMaker, we applied threshold values between 0 and 200 to filter out low-quality fragments. To local similarities identified by BLASTN and TBLASTX, respectively, we applied varying threshold values with respect to the *scores* (measured in bits) that BLAST returns.

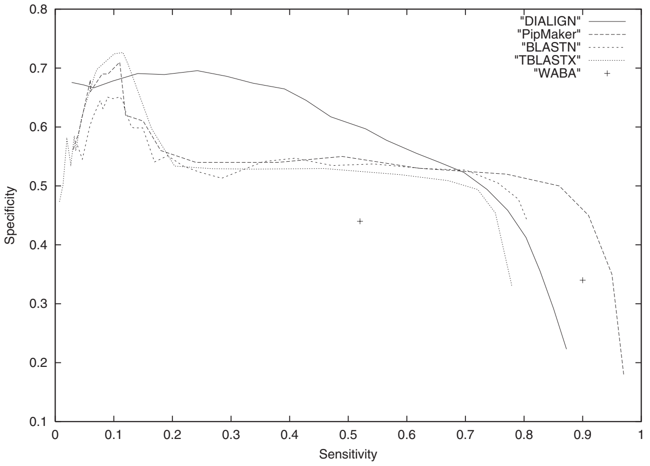

Fig. 3. Specificity-sensitivity plot for DIALIGN, PipMaker, WABA, BLASTN and TBLASTX applied to 42 human–mouse sequence pairs compiled by Jareborg *et al.* (1999). Similarities identified by these programs were compared to annotated exons. DIALIGN was applied with the new iterative scoring scheme (three iteration steps) and both *P-fragments* and *N-fragments* were considered. For DIALIGN and PipMaker, WABA and BLAST, a wide variety of threshold values was applied to the segment pairs returned by these programs as described in the *results* section. In this example, all three programs have high *sensitivity* values since all genes in the aligned human sequences have close homologues in the corresponding mouse sequences. By contrast, *specificity* is lower than in the *Arabidopsis*–tomato example (Figure 4) since, because of the relatively small evolutionary distance between human and rodent, there is still considerable sequence conservation in the *non-coding* regions of the sequences, compare also Figure 5.

The results of these test runs are summarized in Tables 1 and 2, specificity of DIALIGN, PipMaker, WABA and BLAST are plotted against sensitivity with various cut-off values in Figures 3 and 4 and graphical representations of two DIALIGN alignments are shown in Figures 5 and 6. As expected, DIALIGN was more sensitive but less specific if the iterative procedure was applied and, similarly, sensitivity was increased at the expense of specificity if both P-fragments and N-fragments were considered. For the human–mouse test examples, the correlation between fragments returned by DIALIGN and annotated exons was best if only one iteration step was performed but both P-fragments and N-fragments were considered (approximate correlation = 0.57). This was among the highest *AC* value for the five programs if no cut-off was applied to the resulting alignments (BLASTN performed slightly better with an *AC* value of 0.58). Here, DIALIGN was the most *specific* alignment method ($Sp = 0.44$) while PipMaker was the most *sensitive* method ($Sn = 0.97$). For DIALIGN and PipMaker, specificity could be considerably increased by applying a threshold to the fragment scores. A moderate threshold could also increase the *approximate correlation* value; the highest *AC* value was obtained by PipMaker with a threshold of 50 ($AC = 0.65$). Further increased threshold values, however, resulted in *decreased* approximate correlation.

In our *Arabidopsis*–tomato example, all five alignment programs were *more specific* but *less sensitive* than in the human–mouse example. The general tendency, however, was similar for both types of test data in that DIALIGN was most *specific* and PipMaker was the most *sensitive* of the three long-range alignment programs. In this example, however, BLASTX was even more sensitive than PipMaker. In the *Arabidopsis*–tomato

Table 1. Performance of DIALIGN, PipMaker, WABA and BLAST in view of their ability to detect protein-coding regions in large genomic sequences. Test data are 42 human–mouse sequence pairs (Jareborg *et al.*, 1999). Aligned residues are counted as *true positives* (TP) if they are part of annotated exons and as *false positives* (FP) if they are not; non-aligned residues are *false negatives* (FN) if they belong to exons and *true negatives* (TN) otherwise. Standard measures for prediction accuracy are used, namely, *sensitivity* $Sn = TP/(TP + FN)$, *specificity* $Sp = TP/(TP + FP)$, and *approximate correlation* $AC = 0.5((TP/(TP + FN)) + (TP/(TP + FP) + (TN/(TN + FP) + (TN/(TN + FN)) - 1$. Various threshold values have been applied to fragments (segment pairs) identified by DIALIGN and PipMaker as described in the *results* section.

	Human–Mouse						
	TP	FP	TN	FN	Sn	Sp	AC
DIALIGN, all fragments., three iteration steps							
	88 570	308 618	1 281 748	12 956	0.87	0.22	0.44
$w > 10$	78 386	92 564	1 497 802	23 140	0.77	0.45	0.57
$w > 20$	57 411	42 033	1 548 333	44 115	0.56	0.57	0.54
$w > 30$	34 254	16 545	1 573 821	67 272	0.33	0.67	0.48
DIALIGN all fragments one iteration step							
	85 078	128 951	1 461 415	16 448	0.83	0.39	0.57
$w > 10$	77 927	81 148	1 509 218	23 599	0.76	0.48	0.59
$w > 20$	57 411	42 033	1 548 333	44 115	0.56	0.57	0.54
$w > 30$	34 254	16 545	1 573 821	67 272	0.33	0.67	0.48
DIALIGN P-fragments only three iteration steps							
	69 068	128 260	1 462 106	32 458	0.68	0.35	0.46
$w > 10$	65 128	63 899	1 526 467	36 398	0.64	0.50	0.54
$w > 20$	55 193	39 532	1 550 834	46 333	0.54	0.58	0.53
$w > 30$	34 254	16 545	1 573 821	67 272	0.33	0.67	0.48
DIALIGN P-fragments only one iteration step							
	68 263	84 299	1 506 067	33 263	0.67	0.44	0.52
$w > 10$	65 077	61 865	1 528 501	36 449	0.64	0.51	0.54
$w > 20$	55 193	39 532	1 550 834	46 333	0.54	0.58	0.53
$w > 30$	34 254	16 545	1 573 821	67 272	0.33	0.67	0.48
PipMaker							
	98 924	440 852	1 149 514	2 602	0.97	0.18	0.43
$sc > 50$	90 524	97 124	1 493 242	11 002	0.89	0.48	0.65
$sc > 100$	63 356	54 819	1 535 547	38 170	0.62	0.53	0.55
$sc > 150$	34 746	28 840	1 561 526	66 780	0.34	0.54	0.41
$sc > 200$	18 894	14 313	1 576 053	82 632	0.18	0.56	0.34
WABA							
C	53 107	65 246	1 525 120	48 419	0.52	0.44	0.45
C + H	91 379	176 672	1 413 694	10 147	0.90	0.34	0.56
BLASTN							
	81 648	102 414	1 487 952	19 878	0.80	0.44	0.58
$sc > 50$	78 717	81 271	1 509 095	22 809	0.77	0.49	0.60
$sc > 100$	63 823	56 724	1 533 642	37 703	0.62	0.52	0.54
TBLASTX							
	79 073	159 909	1 430 457	22 453	0.77	0.33	0.49
$sc > 50$	74 860	81 478	1 508 888	26 666	0.73	0.47	0.57
$sc > 100$	59 716	55 302	1 535 064	41 810	0.58	0.51	0.52

alignment, DIALIGN reached higher *AC* scores than all other programs, but this time the highest *AC* value was obtained by considering *P-fragments* returned during all three iteration steps.

It should be mentioned that DIALIGN is considerably slower than the other long-range alignment programs in our study (and, of course, far slower than BLAST). For example, WABA aligned a pair of human and murine sequences of 23.8 kb and 19.7 kb, respectively, in 6 minutes and 12 seconds on a Pentium III (451 MHz) under Linux while DIALIGN took 21 minutes and 49 seconds for the same data set on the same machine. A direct comparison of these results with PipMaker is difficult since PipMaker is only accessible through a

Table 2. Performance of DIALIGN, PipMaker, WABA and BLAST on a pair of genomic sequences from *A. thaliana* and tomato. Abbreviations are as in Table 1

			Arabidopsis–Tomato				
	TP	FP	TN	FN	Sn	Sp	AC
			DIALIGN, all fragments, three iteration steps				
	4 622	1 005	19 967	6 406	0.41	0.82	0.47
$w > 5$	3 844	174	20 798	7 184	0.34	0.95	0.52
$w > 10$	3 293	31	20 941	7 735	0.29	0.99	0.50
$w > 15$	2 077	20	20 952	8 951	0.18	0.99	0.43
			DIALIGN, all fragments, one iteration step				
	3 787	26	20 946	7 241	0.34	0.99	0.53
$w > 5$	3 533	22	20 950	7 495	0.32	0.99	0.52
$w > 10$	3 215	22	20 950	7 813	0.29	0.99	0.50
$w > 15$	2 077	20	20 952	8 951	0.18	0.99	0.43
			DIALIGN, P-fragments only, three iteration steps				
	4 317	204	20 768	6 711	0.39	0.95	0.54
$w > 5$	3 719	31	20 941	7 309	0.33	0.99	0.53
$w > 10$	3 293	31	20 941	7 735	0.29	0.99	0.50
$w > 15$	2 077	20	20 952	8 951	0.18	0.99	0.43
			DIALIGN, P-fragments only, one iteration step				
	3 761	22	20 950	7 267	0.34	0.99	0.53
$w > 5$	3 533	22	20 950	7 495	0.34	0.99	0.52
$w > 10$	3 215	22	20 950	7 813	0.29	0.99	0.50
$w > 15$	2 077	20	20 952	8 951	0.18	0.99	0.43
			PipMaker				
	4 938	1 059	19 913	6 090	0.44	0.82	0.49
$sc > 50$	3 051	43	20 929	7 977	0.27	0.98	0.49
$sc > 100$	1 137	21	20 951	9 891	0.10	0.98	0.38
$sc > 150$	679	6	20 966	10 349	0.06	0.99	0.36
			WABA				
C	3 973	144	20 828	7 055	0.36	0.96	0.53
C + H	4 282	308	20 664	6 746	0.38	0.93	0.53
			BLASTN				
	1 374	215	20 757	9 654	0.12	0.86	0.33
$sc > 50$	1 030	20	20 952	9 998	0.09	0.93	0.37
$sc > 100$	510	8	20 964	10 518	0.04	0.98	0.34
			TBLASTX				
	6 135	1 982	18 990	4 893	0.55	0.75	0.50
$sc > 50$	5 191	240	20 732	5 837	0.47	0.95	0.59
$sc > 100$	3 555	60	20 912	7 473	0.32	0.98	0.52

WWW interface (http://bio.cse.psu.edu/pipmaker/) but the PipMaker WWW server returned an alignment of the above sequences by e-mail in a matter of seconds.

DISCUSSION

Comparative analysis of large genomic sequences is an efficient and increasingly important way of predicting functional elements such as protein-coding regions or regulatory elements. Recently, a number of gene-finding algorithms have been proposed that are based on alignments of syntenic genomic sequences (Bafna and Huson, 2000; Batzoglou *et al.*, 2000; Korf *et al.*, 2001; Wiehe *et al.*, 2001; Novichkov *et al.*, 2001). While standard gene-prediction programs either require detailed statistical knowledge about the genes to be detected or depend on the presence of closely related known genes or proteins in databases, the main advantage of the new comparative methods is that they are based on simple measures of local sequence similarity. They are therefore generally applicable without requiring any species-specific training data—provided syntenic sequences from related organisms are available. As all comparative sequence-analysis

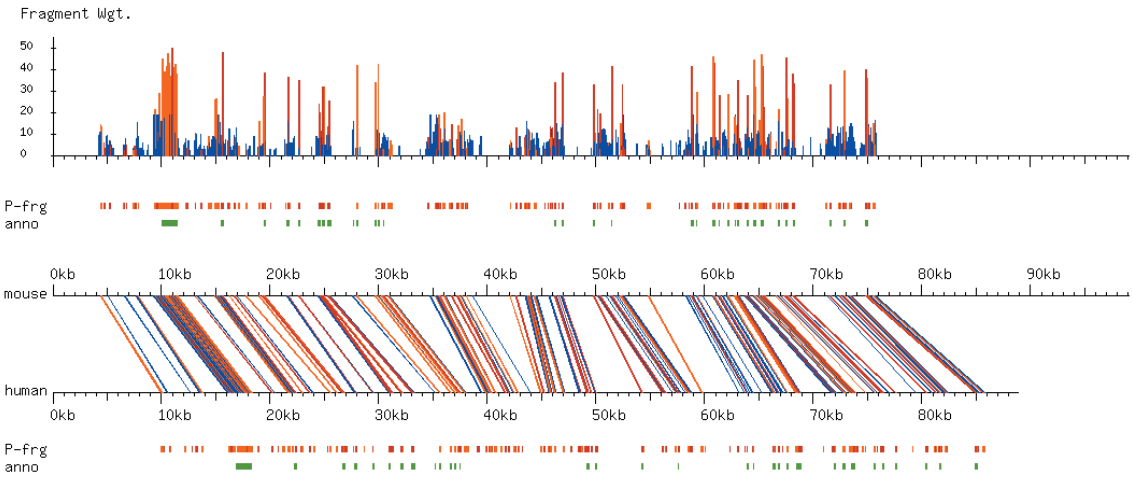

Fig. 5. Visualization of an alignment of human and murine genomic sequences as produced by DIALIGN. Lines connecting the sequences represent fragments (segment pairs) selected in the first iteration step of the alignment procedure. Blue lines are fragments selected in view of their similarity at the *nucleotide level* (N-fragments) while red and orange lines are fragments with higher similarity at the *peptide level* (P-fragments) at the *plus strand* (red) and at the *minus strand* strand (orange), respectively. Vertical bars above the sequences represent the *weight scores* of the selected fragments (`Fragment Wgt.`). Annotated exons are shown in green (`anno`); positions of the P-fragments are indicated by red and orange bars above or below the annotated exons in order to make a direct comparison possible. Low-complexity regions and repeats have been masked using the RepeatMasker software prior to alignment.

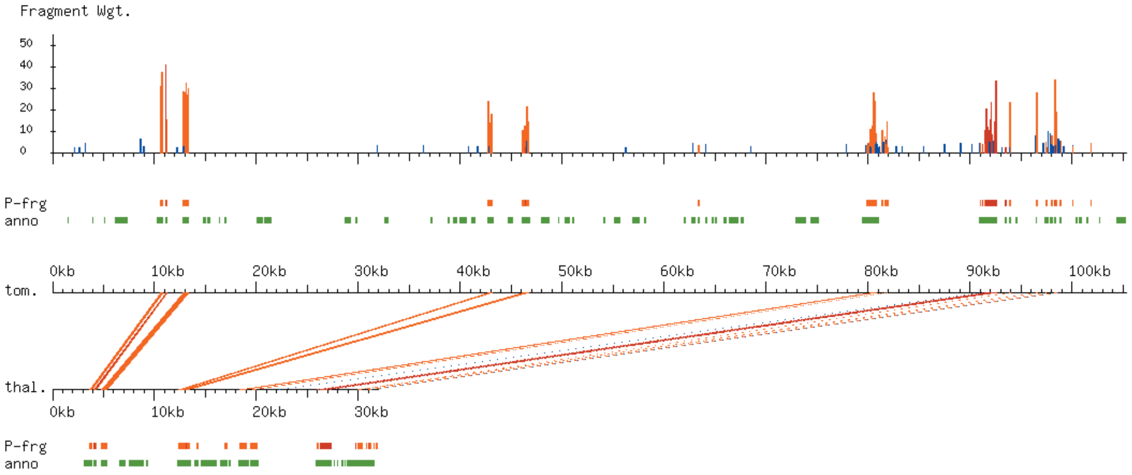

Fig. 6. Alignment of genomic sequences from *A. thaliana* and tomato as produced by DIALIGN. Color coding and abbreviations are as in Figure 5.

methods, homology-based gene-finding approaches critically depend on sensitivity and specificity of the underlying alignment tools.

In this paper, we described a new version of the alignment program *DIALIGN* and we evaluated this program together with four alternative alignment tools, *PipMaker*, *WABA*, *BLASTN* and *TBLASTX*. Our results show that local sequence similarities identified by these tools are highly correlated to protein-coding exons, see for example Figures 5 and 6 for a comparison of the DIALIGN results with annotated exons. It should be emphasized that all alignment programs were run with default parameters, i.e. they did not use any species-dependent information and the human-mouse alignments were done with exactly the same parameter settings as the *Thaliana*–tomato alignment. Varying cut-off values were applied to exclude low-scoring regions from the alignments produced by DIALIGN, PipMaker and BLAST. An important result of

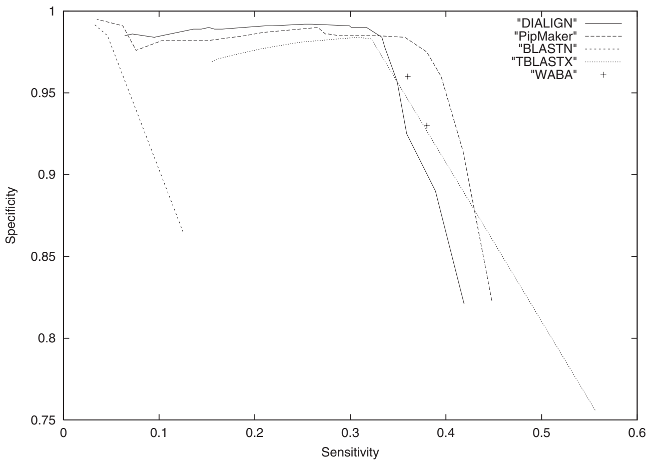

Fig. 4. Specificity–sensitivity plot for different alignment programs applied to a pair of large genomic sequences from *A. thaliana* and tomato. Thresholds and parameter settings are as in Figure 3, a graphical representation of the DIALIGN alignment of these sequences is shown in Figure 6. We have deliberately chosen a sequence pair with little overall similarity so some of the genes are contained in only one of the two sequences and can therefore not be detected by aligning this sequence pair. Consequently, all three alignment programs have low *sensitivity* values. By contrast, their *specificity* is much higher than in the human-mouse example (Figure 3 and 5) since among the *Arabidopsis*–tomato test sequences there seems to be little sequence conservation outside the protein-coding regions.

these experiments is that, among the long-range alignment programs that we have tested, PipMaker is the most *sensitive* method while DIALIGN is most *specific*. Figures 3 and 4 demonstrate that, for a wide range of parameters, DIALIGN is more specific than all other programs in our study while having the same sensitivity.

With the *mixed-alignment* option introduced in this paper, DIALIGN can directly compare two distinct levels of sequences similarity, namely similarity at the *nucleotide level* and at the *peptide level*. Sensitivity to peptide-level similarity is particularly important where sequences from distantly related species are compared and a large proportion of *synonymous* substitutions have occurred. Rivas and Eddy recently proposed a HMM-based algorithm that is able to distinguish these different levels of sequence similarity in a *given* alignment in order to discriminate between protein-coding genes and non-coding RNA genes (Rivas and Eddy, 2001). In contrast, our approach considers nucleotide-level and peptide-level similarity to *construct* alignments of genomic sequences that may be locally related at both respective levels.

Comparison of the BLASTN and TBLASTX results demonstrates that, for closely related species, sequence comparison at the *nucleotide level* can give better results than comparison at the *peptide level* while, for larger evolutionary distances, analysis at the peptide level is superior; this result is in accordance with similar observations by Wiehe *et al.* (2000). To our knowledge, DIALIGN and WABA are currently the only available tools that consider these two types of similarity in the process of sequence alignment. This enables these methods to identify conserved protein-coding regions in unaligned sequences of distantly related species by their similarity at the *peptide level* while, in the same alignments, conserved non-coding regions can be detected by their similarity at the *nucleotide level*. Recent results on local sequence conservation in non-coding DNA suggest that the latter similarities may correspond to non-coding functional sites such as regulatory elements (Loots *et al.*, 2000; Wasserman *et al.*, 2000; Göttgens *et al.*, 2000, 2001). The iterative scoring scheme introduced in this paper makes it possible to detect highly conserved regions in a first iteration step while in subsequent steps, weaker similarities that may correspond to smaller functional elements can be identified.

Long-range sequence alignments can reveal homologies provided they occur in the same relative order within the input sequences. This restriction limits the number of local similarities that can be represented in one single alignment; it works as a *filter* that can greatly reduce the noise generated by spurious sequence similarities. Order-preserving alignment methods can therefore detect even weak local homologies that would seem insignificant if they were found isolated in the input sequences—as long as they are co-linear with other homologies. For this reason, for example, PipMaker uses a lower threshold value and is therefore more *sensitive* if the *chaining option* is used that respects the order of local similarities and returns an optimal *chain* of local alignments. It is well known that large-scale genome rearrangements occur relatively rarely during evolution so, even for distantly related species, gene order is conserved within large parts of the genome sequence. Consequently, order-preserving long-range alignment methods are generally superior to alternative approaches that return collections of local sequence similarities regardless of their relative order.

If distantly related species are compared, the colinearity requirement is, however, a certain limitation as functional sites occurring in different order in the input sequences will necessarily be missed. In Figure 6, for example, the tomato sequence contains large insertions relative to its counterpart from *Arabidopsis* and vice versa. Since exons cannot be detected by aligning these two regions, DIALIGN, PipMaker, WABA and BLAST had relative low *sensitivity* and *approximate correlation* values. In the tomato–*Arabidopsis* example, on the other hand, all alignment methods that we tested were highly *specific* since the evolutionary distance between is relatively large and there seems to be little conservation in the *non-functional* parts of the sequences. The situation is different with the human–mouse sequence pairs that we have used as our first test data. Here, gene order and orientation are largely conserved so most exons could be detected by sequence alignment. In this example, the problem was rather that—because of the relatively small evolutionary distance between primates and rodents—considerable stretches of non-coding sequence are conserved as well. Consequently, all alignment methods were highly *sensitive* but less *specific* than in the tomato–*Thaliana* example.

Generally, the results of any comparative sequence analysis method crucially depend on the evolutionary distance between the analyzed sequences. The distance should be large enough to ensure that conserved functional elements can be distinguished from randomly mutating 'junk' DNA. However, since only those functional sites can be detected that appear in the same relative order and orientation, the compared sequences should not be too far apart. With the growing amount of genome data that are available in public databases, syntenic sequences at varying evolutionary distances can be identified and compared and future studies will show which distances are most appropriate to detect exons and other functional elements.

ACKNOWLEDGEMENTS

We would like to thank Dr. Webb Miller and two unknown referees for their helpful comments on various versions of the manuscript and Dr. Stefan Kurtz, Dr. Enno Ohlebusch, Michael Hoehl and Mike Brudno for their help with REPuter and CHAOS. B.M. was supported by BMBF grant FKZ 01KW9703.

REFERENCES

Abdeddaïm,S. and Morgenstern,B. (2001) Speeding up the DIALIGN multiple alignment program byusing the 'greedy alignment of biological sequences library'(GABIOS-LIB). *Lecture Notes in Computer Science*, **2066**, 1–11.

Altschul,S.F., Gish,W., Miller,W., Myers,E.M. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ansari-Lari,M.A., Oeltjen,J.C., Schwartz,S., Zhang,Z., Muzny,D.M., Lu,J., Gorrell,J.H., Chinault,A.C., Belmont,J.W., Miller,W. and Gibbs,R.A. (1998) Comparative sequence analysis of a gene-rich cluster at human chromosome 12p13 and its syntenic region in mouse chromosome 6. *Genome Res.*, **8**, 29–40.

Bafna,V. and Huson,D.H. (2000) The conserved exon method for gene finding. In Altmann,R., Bailey,T., Bourne,P., Gribskov,M., Lengauer,T., Shindyalov,I., Eyck,L.T. and Weissig,H. (eds), *Proceedings of the 8th International Conference onIntelligent Systems for Molecular Biology*. AAAI Press, Menlo Parc, CA, pp. 3–12.

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.

Birney,E. and Durbin,R. (2000) Using GeneWise in the *Drosophila* annotation experiment. *Genome Res.*, **10**, 547–548.

Brudno,M. and Morgenstern,B. CHAOS: A Heuristic Algorithm for Local Alignment. *Pacific Symposium on Biocomputing* 2002, poster presentation, http://www.stanford.edu/∼brudno/chaos/

Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.

Burge,C. and Karlin,S. (1998) Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.*, **8**, 346–354.

Burset,M. and Guigó,R. (1996) Evaluation of gene structure prediction programs. *Genomics*, **34**, 353–367.

Claverie,J.-M. (1997) Computational methods for the identification of genes in vertebrate genomic sequences. *Hum. Mol. Genet.*, **6**, 1735–1744.

Delcher,A.L., Kasif,S., Fleischmann,R.D., Peterson,J., White,O. and Salzberg,S.L. (1999) Alignment of whole genomes. *Nucleic Acids Res.*, **27**, 2369–2376.

Frishman,D., Mironov,A., Mewes,H.-W. and Gelfand,M. (1998) Combining diverse evidence for gene recognition in completely sequenced bacterial genomes. *Nucleic Acids Res.*, **26**, 2941–2947.

Gelfand,M.S., Mironov,A.A. and Pevzner,P.A. (1996) Gene recognition via spliced sequence alignment. *Proc. Natl Acad. Sci. USA*, **93**, 9061–9066.

Gish,W. and States,D. (1993) Identification of protein coding regions by database similarity search. *Nature Genet.*, **3**, 266–272.

Göttgens,B., Barton,L., Gilbert,J., Bench,A., Sanchez,M., Bahn,S., Mistry,S., Grafham,D., McMurray,A., Vaudin,M. *et al.* (2000) Analysis of vertebrate SCL loci identifies conserved enhancers. *Nature Biotechnol.*, **18**, 181–186.

Göttgens,B., Gilbert,J., Barton,L., Grafham,D., Rogers,J., Bentley,D. and Green,A. (2001) Long-range comparison of human and mouse SCL loci: localized regions of sensitivity to restriction endonucleases correspond precisely with peaks of conserved noncoding sequences. *Genome Res.*, **11**, 87–97.

Hardison,R.C., Oeltjen,J. and Miller,W. (1997) Long human-mouse sequence alignments reveal novel regulatory elements: a reason to sequence the mouse genome. *Genome Res.*, **7**, 959–966.

Henikoff,S. and Henikoff,J. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Jang,W., Hua,A., Spilson,S.V., Miller,W., Roe,B.A. and Meisler,M.H. (1999) Comparative sequence of human and mouse BAC clones from the *mnd2* region of chromosome 2p13. *Genome Res.*, **9**, 53–61.

Jareborg,N., Birney,E. and Durbin,R. (1999) Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Res.*, **9**, 815–824.

Kent,W.J. and Zahler,A.M. (2000) Conservation, regulation, synteny, and introns in a large-scale *C. briggsae–C. elegans* genomic alignment. *Genome Res.*, **10**, 1115–1125.

Korf,I., Flicek,P., Duan,D. and Brent,M.R. (2001) Integrating genomic homology into gene structure prediction. *Bioinformatics*, (17), S140–S148.

Krogh,A., Mian,I. and Haussler,D. (1994) A Hidden Markov Model that finds genes in *E. coli* DNA. *Nucleic Acids Res.*, **22**, 4768–4778.

Ku,H.-M., Vision,T., Liu,J. and Tanksley,S.D. (2000) Comparing sequenced segments of the tomato and *Arabidopsis* genomes: Large-scale duplication followed by selective gene loss creates a network of synteny. *Proc. Natl Acad. Sci. USA*, **97**, 9121–9126.

Kurtz,S., Ohlebusch,E., Schleiermacher,D., Stoye,J. and Giegerich,R. (2000) Computation and visualization of degenerate repeats in complete genomes. *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Parc, CA, pp. 228–238.

Kurtz,S. and Schleiermacher,C. (1999) REPuter: Fast computation of maximal repeats in complete genomes. *Bioinformatics*, **15**, 426–427.

Loots,G.G., Locksley,R.M., Blankespoor,C.M., Wang,Z.E., Miller,W., Rubin,E.M. and Frazer,K.A. (2000) Identification of a coordinate regulator of interleukins 4, 13, and 5 by cross-species sequence comparisons. *Science*, **288**, 136–140.

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

Mallon,A.M., Platzer,M., Bate,R., Gloeckner,G., Botcherby,M.R., Nordsiek,G., Strivens,M.A., Kioschis,P., Dangel,A., Cunningham,D. *et al.* (2000) Comparative genome sequence analysis of the Bpa/Str region in mouse and man. *Genome Res.*, **10**, 758–775.

Miller,W. (2000) So many genomes, so little time. *Nature Biotechnol.*, **18**, 148–149.

Miller,W. (2001) Comparison of genomic DNA sequences: solved and unsolved problems. *Bioinformatics*, **17**, 391–397.

Morgenstern,B. (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, **15**, 211–218.

Morgenstern,B. (2000) A space-efficient algorithm for aligning large genomic sequences. *Bioinformatics*, **16**, 948–949.

Morgenstern,B. and Atchley,W.R. (2001) Evolution of bHLH transcription factors: modular evolution by domain shuffling? *Mol. Biol. Evol.*, **16**, 1654–1663.

Morgenstern,B., Dress,A.W. M. and Werner,T. (1996) Multiple DNA and protein sequence alignment based on segment-to-segment comparison. *Proc. Natl Acad. Sci. USA*, **93**, 12098–12103.

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.

Novichkov,P.S., Gelfand,M.S. and Mironov,A.A. (2001) Gene recognition in eukaryotic DNA by comparison of genomic sequences. *Bioinformatics*, **17**, 1011–1018.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Rinner,O. and Morgenstern,B. (2001) Gene prediction by comparative sequence analysis. *Proceedings of the German Conference on Bioinformatics*. pp. 131–134.

Rivas,E. and Eddy,S. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, 2:8.

Schwartz,S., Zhang,Z., Frazer,K.A., Smit,A., Riemer,C., Bouck,J., Gibbs,R., Hardison,R. and Miller,W. (2000) PipMaker-a web server for aligning two genomic DNA sequences. *Genome Res.*, **10**, 577–586.

Smith,T.F. and Waterman,M.S. (1981) Comparison of biosequences. *Advances in Applied Mathematics*, **2**, 482–489.

Stormo,G.D. (2000) Gene-finding approaches for eukaryotes. *Genome Res.*, **10**, 394–397.

The Arabidopsis Genome Initiative (2000) Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. *Nature*, **408**, 796–815.

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.

Usuka,J. and Brendel,V. (2000) Gene structure prediction by spliced alignment of genomic DNA with protein sequences: increased accuracy by differential splice site scoring. *J. Mol. Biol.*, **297**, 1075–1085.

Wasserman,W., Palumbo,M., Thompson,W., Fickett,J. and Lawrence,C. (2000) Human-mouse genome comparisons to locate regulatory sites. *Nature Genet.*, **26**, 225–228.

Wiehe,T., Guigó,R. and Miller,W. (2000) Genome sequence comparisons: Hurdles in the fast lane to functional genomics. *Briefings in Bioinformatics*, **1**, 381–388.

Wiehe,T., Gebauer-Jung,S., Mitchell-Olds,T. and Guigo,R. (2001) SGP-1: Prediction and validation of homologous genes based on sequence alignments. *Genome Res.*, **11**, 1574–1583.

Zhang,Z., Berman,P. and Miller,W. (1998) Alignments without low-scoring regions. *J. Comput. Biology*, **5**, 197–210.

Zhang,Z., Raghavachari,B., Hardison,R. and Miller,W. (1994) Chaining multiple-alignment blocks. *J. Comput. Biology*, **1**, 217–226.