

# **Examining the Current Problems of Whole Genome Comparison: A Review**

**Biochemistry 218 Project**

**Patrick Chain**  
[chain2@llnl.gov](mailto:chain2@llnl.gov)

## **Abstract**

With the continuing improvements in high-throughput genomic sequencing and the ever-expanding sequence databases, new advances in software programs for post-sequencing functional analysis are being demanded by the general scientific community. Whole genome comparisons have been heralded as the next logical step toward solving genomic puzzles, such as determining coding regions, discovering regulatory signals, and deducing the mechanisms and history of genome evolution. However, before any such detailed analyses can be addressed, methods are required for comparing (alignments) and displaying (visualization tools) such large sequences. These two topics are reviewed herein.

## **Sequencing: Too fast?**

The output of sequence data from world-wide sequencing centers with constantly increasing sequencing capacities has been rising at an exponential rate for the past decade or two (see <http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html>). The first two publications of microbial whole genome sequencing projects were published in 1995. Only six and a half years later, there are almost 60 completed, annotated genomes available (most of them eubacteria and archaeobacteria but also a yeast), along with draft analyses of several multi-cellular eukaryotes such as nematode, fly, man, weed, mouse, rat and fish. More still are currently underway. The increase in sequencing efficiency means that the bottleneck is not the accumulation of raw data but the annotation and analysis of sequences and genomes.

## **Annotation: Not so fast?**

One of the primary goals in analyzing complete genomes is to identify all the functional regions in the sequence, including genes and regulatory regions. Gene finding is relatively straightforward for compact microbial genomes due to very small intergenic regions, whereas the “signal-to-noise” ratio for more complex eukaryotic genomes makes gene prediction extremely difficult. Bacterial genomes consist mostly (85-95 %) of coding sequence, the human genome encodes only ~3%, while a vast array of eukaryotic organisms have coding potentials that lie between these two extremes.

There are two computational strategies for identifying genes: 1) extrinsic methods that take advantage of the repository of known or proposed genes and proteins through database similarity searches (for most of the bacterial genomes, roughly 70 % of the annotated genes of any one genome have homologues in other species), and 2) intrinsic (*ab initio* or *de novo*) methods that use probabilistic Hidden Markov Models to predict protein coding regions (these models incorporate into their decision-making the statistical patterns of nucleotide ordering within encoding regions - genome features such as relative amino acid, codon usage, and dicodon frequencies). These programs include CRITICA (Badger and Olsen 1999), GLIMMER (Salzberg *et al.* 1998, Delcher *et al.* 1999a), GENMARK (Borodovsky *et al.* 1993), GRAIL (Uberbacher and Mural 1991, Xu *et al.* 1994), and GENSCAN (Burge and Karlin 1997).

Automated gene and gene function predictions, although an indispensable requirement for genome sequencing projects, have been the subject of great controversy (Devos and Valencia 2001, Kyripides and Ouzounis 1999, Galperin and Koonin 1998, Brenner 1999, Dandekar *et al.* 2000). For example, only one month after the release of

the *Haemophilus influenzae* genome (Fleischmann *et al.* 1995), 148 amendments to the annotation were published by separate authors (Casari *et al.* 1995). Since these types of false predictions are misleading and tend to be perpetuated to other genomes, appropriate and accurate annotation techniques must not be underscored.

### **Comparative Genomics: To the Rescue**

The potential for cross-species comparison to help reveal conserved coding regions as well as other regions of potential biologic function has only recently become clear. The use of alignment-based comparisons to uncover conserved functional elements has been termed “phylogenetic footprinting” (Tagle *et al.* 1988). Of importance to annotation, this approach obviates the need for *a priori* knowledge of a sequence motif and provides a complement for algorithmic analyses.

It is generally believed that homologous genes are relatively well preserved, while non-coding regions tend to show varying degrees of conservation. Non-coding regions that do show conservation are thought important for regulating gene expression, maintaining the structural organization of the genome and possibly have other, yet unknown functions. Several comparative sequence analysis approaches using alignments have recently been used to analyze corresponding coding and non-coding regions from different species, although mainly between human and mouse (Hardison *et al.* 1997, Lund *et al.* 2000, Batzoglou *et al.* 2000, Kent and Zahler, 2000a, Dubchak *et al.* 2000, Jareborg *et al.* 1999, Stojanovic *et al.* 1999, Gelfand *et al.* 2000). Of course, the utility of cross-species comparative genomics in the identification of such regions is greatly influenced by the evolutionary distance of the species in question.

Comparative analysis of a number of phylogenetically diverse genomes may provide clues about the selective pressures governing gene/operon clustering and may offer insights into mechanisms of evolution or show patterns in acquisition of foreign material via horizontal gene transfer. Genome comparisons of more closely related species may also help determine the genetic basis for phenotypic variation and may reveal species-specific regions (signatures) that can be targeted for identification. Detection techniques based on knowledge of such regions has recently proven fruitful for forensics analysis in the recent anthrax outbreaks.

Although it was once the goal to characterize the genomes of a member from many, if not all, of the distant branches of the phylogenetic tree, it is now becoming more common for a genome-sequencing project to target an organism that is very closely related to an already-sequenced genome. This is reflected in the number of recent publications detailing such comparisons. Indeed, along with the above-mentioned eukaryotic comparative analysis papers, there now exist six publications of bacterial inter- and intra-species whole genome comparisons (Alm *et al.* 1999, Read *et al.* 2000, Hayashi *et al.* 2001, Perna *et al.* 2001, Ogata *et al.* 2001, Glaser *et al.* 2001).

Underlying these genomic comparisons are alignment programs, some of which have been recently developed to tackle the various problems of dealing with long nucleotide strings such as genomes. Only a handful of analysis tools, specifically alignment tools, are available to deal with comparing large sequences such as whole genomes or chromosomes. In addition to this already complex problem is the issue of parsing and reporting/displaying this data, since these alignments and their visualization/interpretation must go hand in hand.

## Alignments: Problems and Progress

Alignment of nucleic or amino acid sequences has been one of the most important tools in sequence analysis, with much research and many sophisticated algorithms available for aligning sequences with similar regions. These require assigning a score to all the possible alignments (typically, the sum of the similarity/identity values for each aligned residue, minus a penalty for the introduction of gaps), along with an algorithm to find optimal or near-optimal alignments according to this scoring scheme. Needleman-Wunsch (1970) and Smith-Waterman (1981) accomplished this using a dynamic programming approach.

Until very recently, most of these algorithms were primarily designed for comparing single protein sequences or DNA sequences containing a single gene or operon. There are several problems associated with aligning long genomic sequences or entire genomes. Most programs are incapable of producing accurate long alignments. Linear-space Smith-Waterman variants are too computationally demanding without specialized hardware (memory-limited), while other approaches are too time-consuming. There is also a typical tradeoff between higher speed and increased sensitivity.

Genome-length comparative alignment tools have usually been designed with a specific goal in mind: some simply aim to find any and/or all similar, or identical stretches of DNA between two genomes; others specifically target coding sequences (such as exons) and exon order conserved between two distant species; still others focus on intergenic and intronic regions to detect conserved regulatory signals. Some of the main problems associated with these goals lie in dealing with rearrangements (e.g. exon shuffling or other non-syntenous regions resulting from intra-molecular recombinations), large insertions or deletions (sequences that share several regions of local similarity separated by unrelated regions), repeated elements (e.g. duplicated genes/operons, transposons, SINES, LINES etc.), tandem repeats, and inherent problems of gene regulatory elements, including their small(ish) size and relative resistance to small insertions/deletions or substitutions. Another subject infrequently addressed for long sequences, and needing much more in-depth exploration, is the issue of multiple alignments.

Some or all of the above-mentioned problems are addressed by a number of new programs discussed further below, such as ASSIRC (Vincens *et al.* 1998), DIALIGN (Morgenstern *et al.* 1998; Morgenstern 1999), DBA (Jareborg *et al.* 1999), MUMmer (Delcher *et al.* 1999b), PipMaker/BlastZ (Schwartz *et al.* 2000), GLASS (Batzoglou *et al.* 2000), WABA (Kent and Zahler, 2000a), and LSH-ALL-PAIRS (Buhler 2001). Several of these programs are modifications of popular local alignment search tools such as BLAST (Altschul *et al.* 1997) and CrossMatch (<http://www.phrap.org/>), which are based on an efficient algorithm first used by Dumas and Ninio (1982), that finds all short exact matches ( $k$ -words above some minimum length) and extends these so-called “seeds” to make larger contiguous matches (with a maximum number of substitutions or gaps).

Several comparative studies of genomes or of large genomic segments are still using older methodologies to solve their particular problem(s). For example, a very recent study by Oggioni and Pozzi (2001) opted to simply parse a BLASTn search to identify clone-specific blocks of sequence in a comparison of three *Streptococcus pneumoniae* serotypes. Another Smith-Waterman-based program, CrossMatch, was used by Lee *et al.* (1998) in a study of conserved sequences in the non-coding regions of the prion protein

gene locus, between three mammalian species (human, mouse and sheep). In yet another recent study by Lund *et al.* (2000), a filtered dot-plot algorithm, using the program lineplot in the CGAT package (<http://inertia.bs.jhmi.edu/roger/CGAT/CGAT.html>), helped compare syntenic human and mouse regions. Perhaps these comparative methods were used because the algorithms and their outputs were well-suited to the particular study (although some of the new alternatives could also achieve the desired results more quickly), or it may be that the newer “long-range” alignment programs have not yet gained wide-spread exposure or acceptance by the general scientific community.

### **Visualizing Data: Not as Easy as it Looks**

Direct output from alignment programs are typically in the form of text files reporting the actual aligned bases or residues. With the large data sets used in comparing genomes, these results are most often not intuitively interpretable. Visualization tools are therefore necessary to cope with the complexities and sheer volume of data, and present it to biologists in a comprehensive and comprehensible manner. Early work on this problem centered on two-dimensional representations called dot-plots (LAD and LAV - Schwartz *et al.* 1991, Dotter - Sonnhammer and Durbin 1995), but the focus has since shifted to more compact, linear representations (Duret *et al.* 1996, Galili *et al.* 1997).

Similar to alignment algorithms, the direction in development of new viewing/display tools often follows the goals of the research in question. In addition to an interpretable alignment, visualization and browsing tools need to incorporate extra analyses and features such as database homologies and gene predictions from various sources. The ability to locate repetitive elements, alternate start and splice sites, protein binding sites, and other genomic features can help the biologist in his analyses. Interactive features are other useful options to consider, such as the viewing resolution (a static graphic vs. the ability to zoom) and real-time analysis capabilities (e.g. ability to search specific regions for homologies). Other problems include: how to represent breaks in synteny (such as genome rearrangements) if at all, will alignments from both strands be displayed, can and how will multiple alignments be shown, is only one sequence the reference for the alignment(s), and how will contigs be displayed if a non-finished genome is used as one of the entries for the alignment? A further problem lies with the input of data for the visualization programs, since most of these were developed to work on only one specific file format. Gottgens *et al.* (2001) also raised the issue of availability of such software, as some unnamed programs that generated figures shown in publications (Lee *et al.* 1998, Delcher *et al.* 1999b) are not available.

Several of these issues are addressed by a number of recent developments in comparative genome alignment visualization programs such as PipMaker and the Enteric/Menteric/Maj suite (Florea *et al.* 2000, McClelland *et al.* 2000) which are based on PipMaker-like PIPs, Alfresco (Jareborg and Durbin 2000), Intronerator (Kent and Zahler 2000b), VISTA (Dubchak *et al.* 2000, Mayor *et al.* 2000), SynPlot (Gottgens *et al.* 2001), and ACT (<http://www.sanger.ac.uk/Software/ACT/>).

Much of the work to improve the fledgling field of whole genome comparison involves the design of new alignment algorithms and the modification or implementation of existing algorithms. These programs have often been coupled to visualization tools that try to make a seamless transition from raw data to interpretable comparisons. As recently discussed in a review of genomic DNA sequence comparisons by Miller (2001),

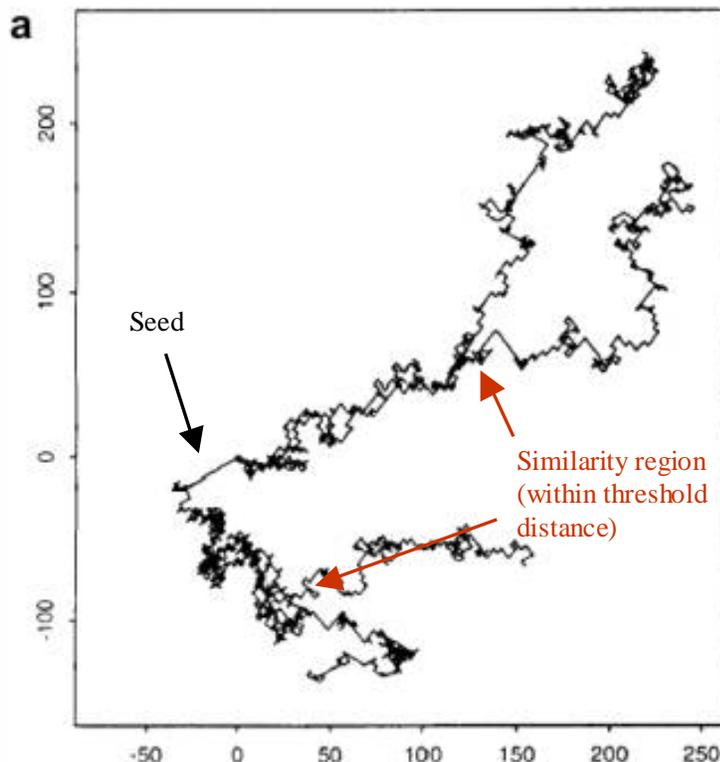
there remains much room for improvement in terms of long-sequence (or whole-genome) alignment algorithms (especially true for multiple alignments), and in terms of formatted or processed graphical output that a user may be able to interpret and combine with other analyses.

### The Progression of Genome Sequence Alignment Programs

As discussed previously, almost all available alignment programs before 1998 were developed to find target regions similar to a single “small” sequence. Because a global alignment strategy for diverged genomic sequences is likely predestined to failure for having to align non-syntenous and unrelated regions in an end-to-end approach, local alignments have been the strategy of choice.

### ASSIRC – Accelerated Search for Similarity Regions in Chromosomes

Vincens *et al.* (1998) developed a tool, called ASSIRC, to find regions of similarity in pair-wise genomic sequence alignments. ASSIRC invokes three steps. 1) Pairs of identical  $k$ -mers (of fixed size  $k$ ), called “seeds”, are identified using standard hashing functions. 2) All seeds are extended using a random walk procedure (the four bases are each associated with a different displacement vector), where the sequences are converted to a two-dimensional graph (Figure 1) and the proximity along the length of the alignment of the two regions are quantified. 3) These regions of similarity are then aligned using standard Smith-Waterman variants; in this study, the BESTFIT program was used for the actual alignment, however other programs may be better suited for aligning larger regions. Although this novel approach proved to be faster and finds more regions not detected by BLAST or FASTA, this algorithm is rather sensitive to large insertions or deletions and does not have a visualization tool associated with it.



**Figure 1:** Graph representing the “random walk” of two subsequences within a region of similarity. (adapted from Vincens *et al.* 1998)

## **DIALIGN – Diagonal ALIGNment**

At the same time, Morgenstern *et al.* (1998) had developed a more versatile alignment program called DIALIGN, capable of both pair-wise and multiple alignments. One of the novelties of this program was the use of gap-free whole segments for comparison, instead of using single bases or residues. The alignments are thus composed of gap-free segments of equal length that would form diagonals in a dot-matrix comparison. Once these diagonals are found (within a set threshold), a scoring function is applied and the collection of segments with the maximum sum of scores (overall optimal alignment) is found by a modification of the standard dynamic programming scheme (for pair-wise alignments). For multiple alignments, the diagonals are first sorted according to both their weight scores and the degree of overlap with other diagonals (overlap weights). The diagonals are then aligned using a “greedy algorithm” (reviewed by Zhang *et al.* 2000), where the segments with highest scores are selected in turn, evaluated for consistency and added to the growing multiple alignment. When no additional diagonal can be incorporated, gaps are then introduced to properly arrange the sequences.

The use of complete segments of sequences in the comparison allows DIALIGN to locate small conserved regions that cannot be detected by standard Smith-Waterman alignment programs which use gap and gap extension penalties (i.e. can identify functionally important regions even in large genomic sequences). This particular feature was found useful by Gottgens *et al.* (2000) in a long-range comparison of the mouse and human *SCL* loci. In their study, a visualization tool called SynPlot was designed to display the DIALIGN alignments.

One drawback as it pertains to repeated sequences is that once a diagonal is incorporated into the alignment, it cannot be removed. Also, although this program overcomes many problems of generating global alignments, its utility is restricted to colinear segments. This obviates its use in comparing draft data (in a set of contigs) to complete, large DNA segments or to other draft data.

## **DBA – DNA Block Aligner**

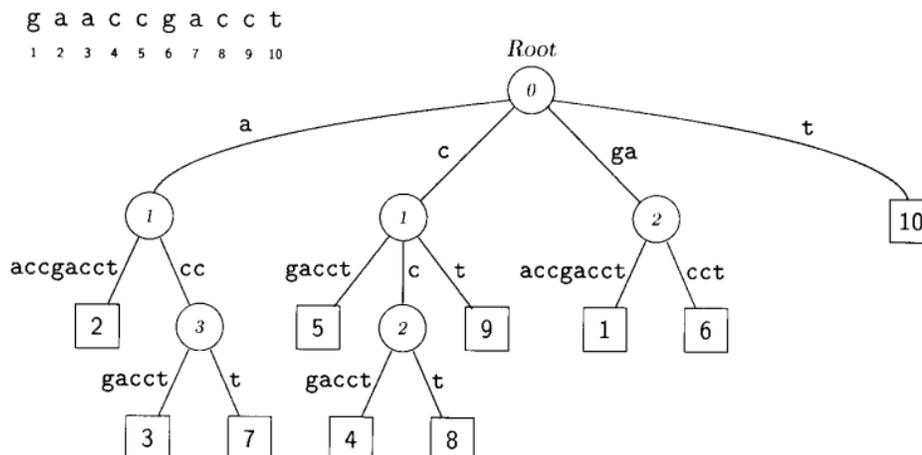
Similar to DIALIGN, another dynamic programming tool for finding conserved, colinear blocks of sequence flanked by non-conserved regions, was reported by Jareborg *et al.* (1999) from Sanger. This program, called DBA, was specifically designed for comparing two eukaryotic non-coding regions, as an alternative to a gapped BLAST program with a complicated post-processing step. These regions can contain small gaps, and may be separated by large gaps. DBA assigns blocks of similarity into four groups based on their levels of conservation (measured by a probabilistic finite state machine, or pair-Hidden Markov Model): 60%-70%, 70%-80%, 80%-90%, and 90%-100%. This pair-HMM is composed of six states: four representing the four blocks of similarity, one as the beginning of a non-matching region, and the other as the end of a non-matching region.

This algorithm has been shown to be more sensitive than BLAST, but has the same shortcomings as DIALIGN: the necessity to mask the repetitive regions before performing an alignment, and the requirement for colinearity in the two aligned sequences (rearrangements or domain shuffling would not be detected). In addition, DBA can only deal with two sequences. DBA alignments can be viewed by a comparative analysis workbench released a year later by Jareborg and Durbin (2000), called Alfresco.

## MUMmer – Maximal Unique Match (mer)

While the previous alignment programs require a pre-screening process to mask the repetitive regions before the initial alignment (to remove spurious matches), the MUMmer algorithm, developed by Delcher *et al.* (1999b) at TIGR, requires no such step. A need to compare closely related bacterial species (strains even) motivated the creation of this pair-wise alignment program, capable of detecting every difference between two microbial genomes. Under the assumption that the compared sequences are closely related, this system can quickly perform high-resolution comparisons of whole genome-length sequences, locating all the single nucleotide polymorphisms (SNPs), insertions/deletions, differences in number and location of repeat elements and tandem repeats, as well as regions repeated in only one of the two sequences. This program is also amenable to detecting the differences between two different versions of a genome sequencing project (two drafts, or a drafted genome vs. a complete one).

Unlike the other algorithms that rely on either dynamic programming or hashing techniques (or both), this program uses a suffix tree approach which can quickly find all the maximal unique matches (MUMs) between two sequences, even for very large inputs. In a sequence, a “suffix” is a subsequence that begins at any position and extends to the end, thus a sequence of length  $N$  has  $N$  suffixes, one starting at each position. In a suffix tree, each path from root node to leaf node represents a unique suffix. Each internal node corresponds to a repeated sequence in the original genome, and the number of times this sequence is repeated equals the number of leaf nodes underneath it (Figure 2). With a suffix tree from one genome, it is possible to add another genome and quickly identify all the MUMs. MUMmer employs a variation of the LIS (longest increasing subsequence) algorithm to consistently order the MUMs to form the basis of an alignment that can span very long “mismatch” regions. With this global alignment done, several methods are used to close the remaining gaps, which consist of SNPs, apparent insertions/deletions (lateral transfer, transpositions), polymorphic regions (closed by a Smith-Waterman approach), and repeated elements, which by the nature of the MUMs (U for Unique) are captured when found out of context compared with the other sequence.



**Figure 2:** Suffix tree for the sequence gaaccgacct. Square nodes are leaves and represent complete suffixes. They are labeled by the starting position of the suffix. Circular nodes represent repeated sequences and are labeled by the length of that sequence. (adapted from Delcher *et al.* 1999)

All three steps (finding MUMs, sorting them, and aligning the remaining gaps) are linear with respect to running time and space and thus the overall time and space are linear, matching other speed- and space-efficient algorithms. The input parameter determining the length of the shortest MUM can be optimized for highly similar genomes or lowered for more distantly related genomes. This system was tested on two bacterial strains, two species of the same genus, and on a 220 kb syntenic regions of mouse and human, and reportedly performed very well (and very rapidly) in all cases. Although a graphical interface was developed, it has not yet been made available, forcing users wanting to scroll along the two genomes to create their own programs to parse and display the data. MUMmer also only performs pair-wise alignments.

### **PipMaker – Percent Identity Plot MAKER**

A web server named PipMaker (Schwartz *et al.* 2000) was first designed to efficiently compare two sequences from 100 to 1000 kb. PipMaker actually serves a dual function, aligning input sequences and displaying them as a percent identity plot or PIP. The underlying high-performance local alignment program, Blastz is a variant of the Gapped BLAST program by Altschul *et al.* (1997) specifically designed for aligning two long sequences.

As previously mentioned, repetitive elements often wreak havoc with these alignment programs, thus most algorithms work better with these regions masked. PipMaker now has two options if one does not want to mask out these regions. When invoking an option called “chaining”, PipMaker removes the confusion by identifying only the matches that appear in the same relative order in the compared sequences. An alternative is the “single coverage” option, which avoids duplicate matches by allowing only the highest scoring set of alignments. Also, PipMaker can compare draft sequence to a single reference sequence (a feature somewhat similar to MUMmer), however the reverse, as well as a draft to draft comparison, is not yet possible, though may soon be made available (Miller 2001). One of the newer features available on the network is the ability to enter multiple sequences, however these sequences are all to be compared to the reference sequence in pair-wise alignments. The main limitation to using PipMaker is that it is only available as a server (restricted to 2 Mb input) and not as a program, thus the results are always displayed as PIPs.

### **GLASS – GLobal Alignment SyStem**

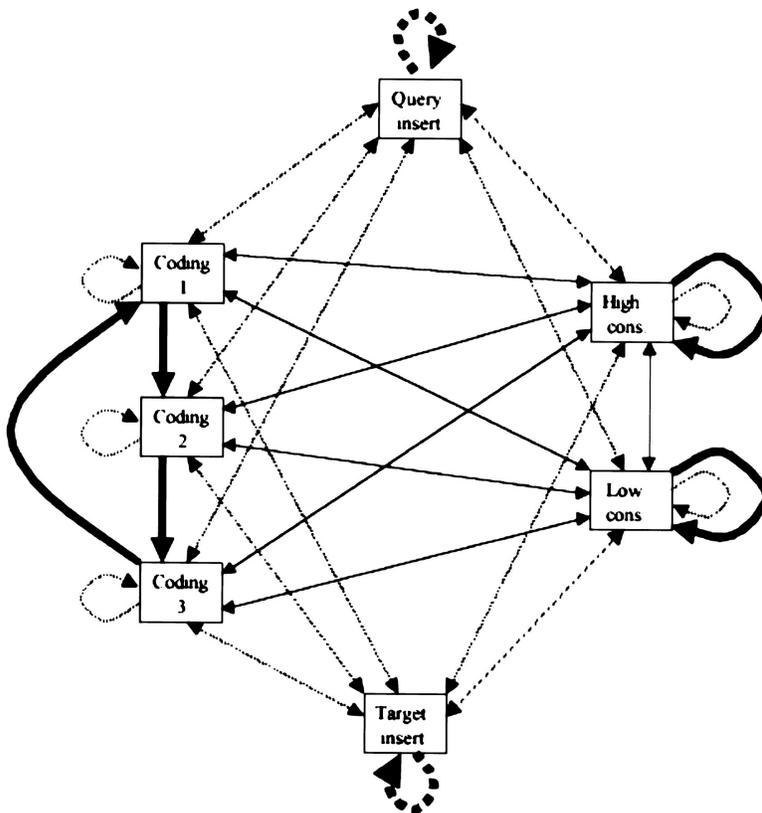
Batzoglou *et al.* (2000) developed another program, GLASS, to perform cross-species (mouse-human) exon recognition. With the idea that some exons may be as small as 50 bp, and flanked by much longer regions of poor(er) similarity, GLASS iteratively finds exact matching seeds (*k*-mers) between two sequences. The genomic sequences being compared are then converted to their subset of seeds. The consistent seeds are given scores based on a standard dynamic programming alignment of the “seed+12 bp on either side” (mismatches and gaps in the alignment receive a score of 0). The regions of good local alignment (passing a threshold) are further evaluated by recursively aligning these regions using smaller seeds. Once complete, the “final” alignments are again extended on either side and the remaining gaps closed using standard dynamic programming. In this study, poorly aligned regions and repeat regions were then masked, such that the remaining conserved sequences could be scanned for genes.

Like DIALIGN, this system easily deals with large insertions between conserved domains, but cannot align non-colinear regions or rearrangements. Although no visualization tool was presented with the program, subsequent research using GLASS did present the modified output with VISTA (Dubchak *et al.* 2000, Mayor *et al.* 2000). Here, the authors used GLASS in a three-way comparative study involving human, mouse and dog, but the alignments were done in pair-wise fashion and intersection/union analyses were performed to find regions conserved in all three sequences.

### WABA – Wobble Aware Bulk Aligner

Near same time as GLASS, Kent and Zahler (2000a) revealed another novel algorithm, called WABA, to research interspecies genomic conservation, using two closely related nematodes (*Caenorhabditis elegans* and *C. briggsae*). Along with accepting small and large insertions/deletions, their three-pass algorithm focused on enabling a good alignment while allowing rapid divergence in the third (wobble) position of codons.

In this process, one genome is decomposed into 8-mers (the positions remain known) where the third and sixth nucleotides are ignored. This 8-mer set is then scanned with the other genome and “hits” are recorded (the position of these 8-mers in both the target and query are noted). If two hits are within 1 kb in both genomes, and their positions indicate they may lie within a homologous region lacking inserts, they are considered promising candidates. These result in clumps of hits which are scored for the best local alignment. Next, a seven-state pair-Hidden Markov Model (Figure 3) is used to align most of this data, producing predictions as to whether a region is coding or not, and to give the alignment a score. Two of these states capture long inserts in the target or query sequence, two others capture highly conserved (~90% base similarity) or lowly conserved (~50% similarity) regions, while three states capture coding regions. The last WABA step simply looks for overlapping alignments with at least 15 identical nucleotides.



**Figure 3:** Pair-wise HMM depicting a seven state aligner. The most likely (thick) and unlikely (thin) transitions out of each state are indicated with arrows. Transitions are associated with aligning pair of bases (solid arrows) or aligning a base with a gap (broken line arrows).

(adapted from Kent and Zahler 2000a)

This is a sensitive pair-wise alignment tool that accounts for divergence in the wobble position of coding regions (the first algorithm to do so), and thus works well to uncover conserved exons and even conserved sequences in syntenous regions (including conserved regulatory elements), but will not reliably align rearrangements or other non-colinear regions. A visualization tool called Intronerator was also designed by the authors (Kent and Zahler 2000b) to aid in interpretation.

### **LSH-ALL-PAIRS – Locality-Sensitive Hashing in ALL PAIRS**

Most recently, Buhler (2001) published a new algorithm, LSH-ALL-PAIRS, to specifically find ungapped alignments in genomic sequences. This program addresses issues in exact seed matching (like in GLASS, PipMaker and ASSIRC) by looking for similar seeds (with a specified fraction of substitutions) using an efficient randomized search technique called locality-sensitive hashing. Exact seed matching requires selecting a minimum seed length that balances sensitivity and weak similarity against efficiency on long sequences to reduce hits by random chance. LSH-ALL-PAIRS is particularly useful for finding similarities with frequent substitutions (including wobble base changes) since the algorithm can find similar sequences using a long seed length (typically 60-80 bp) while allowing several substitutions.

This program is run iteratively to minimize the chance of missing true positives with its random search approach. Like other hashing techniques, the seeds are extended into local alignments (500 bp on either side), helping to recover missed similarities. Overlapping segments are assembled into longer, disjoint ungapped local alignments. These are reported after trimming regions of low similarity at the ends, if they pass a significance threshold.

Some drawbacks are mentioned by the author, even though this algorithm compared well with MUMmer in one specific case. First the gaps between segments may be small and missed in the initial random search. Second, there is no attempt to include gapped alignments, such that long gapped similarities may be missed if their ungapped sub-fragments do not score significantly. Third, the initial seed search was scored simply with a mismatch count instead of a more general alignment scoring function. LSH-ALL-PAIRS, like most of the other algorithms for genomic alignments, does not yet work on multiple alignments. Also, a visualization tool has not yet been addressed for use with this program, and the program itself is not available for use or download.

### **The Progression of Visualization Tools for Displaying Genomic Comparisons**

As well as having been published within the same year, many of the following programs were designed alongside a genomic sequence alignment program, or at least with a particular one in mind. Specific research goals have also played a large part in directing the functionality of these graphic displays. There remains a severe lack of versatile visualization tools to serve the needs of the average molecular biologist, who often has a different research interest, with different display needs.

### **PipMaker and Enteric/Menteric/Maj (displays Blastz alignments)**

As discussed in the previous section, PipMaker performs sequence alignments and outputs the data as a PIP, which is a representation of all the local alignments between two sequences and their qualities (as measured by its percent identity over the

length of the local alignment). The reference sequence is displayed along the horizontal axis, the local alignment matches are horizontal lines within the plot, and their height represents the quality of that match (percent identity). This is meant to display alignments of syntenic regions or to display the presence of reference-sequence counterparts in other sequences, regardless of positioning. The main limitation of this tool is that the positions of the alignments in the other, non-reference sequences are not shown. Thus a particular local alignment may be in a different genomic context in the non-reference sequence, and even in the opposite direction. This graphic is also static, with no zoom capability and no hot-links to other information or databases. However, the results of other analyses can be displayed along the PIP when they are read in as separate files. Another important drawback is that this program is not available for download, and there is a 2 Mb size restriction on the server.

The Enteric/Menteric/Maj suite of programs (Florea *et al.* 2000) uses pre-computed genome-wide pair-wise alignment files generated by PipMaker (in essence, a display database), using *Escherichia coli* as a reference organism (to genomes of several related enteric bacteria such as *Salmonella enterica* and *Yersinia pestis*). An in-depth comparative study of differences between several of these genomes was also reported (McClelland *et al.* 2000). These three programs render a set of PIPs, all in reference to *E. coli*, and several layers of information have been integrated into these applications, such as reports from sequence analysis tools, literature references, locations of known genes, etc. These multiple alignment displays also try to address the colinearity of the aligned regions by highlighting the break in synteny to indicate an edge to a region absent in the reference or absent in the target sequence, however these are still not completely informative. While Enteric (Figure 4 in appendix) and Menteric generate static views of 20 kb and 1 kb regions respectively, Maj provides an interactive graphical display (beginning at one of the same two resolutions) with ability to zoom and to view text alignments of any region within the display. Future versions of Menteric will allow several input file formats, such that other alignment programs (like MUMmer and tFASTx) may feed into this display. The greatest limitation of this suite of tools is that they are only useful for investigating the genomes available on the server, especially *E. coli* since it was used as the reference organism.

### **Alfresco (displays DBA alignments, along with several others)**

The goal in developing Alfresco (Jareborg and Durbin 2000) was to provide an interactive graphic front-end for a variety of analysis programs as they pertained to comparative analysis (Figure 5 in appendix). In addition to a variety of displays (overview graphic, textual alignment and dot-plot), Alfresco can combine alignments with DBA, processed BLASTn results, BLASTx hits, repeats, results from gene modelers (e.g. GENSCAN), expressed sequence tag (EST) hits and CpG islands. Selected regions can be subjected to further analysis (e.g. Dotter). These extra features can be done automatically (in batch mode) and do not have to be entered in manually, unlike PipMaker. Other types of analyses can be incorporated, but would require modification of the source code.

Conserved regions are shown by color and are attached by a line, which suggests an ability to represent non-colinear regions (although DBA does not maintain that functionality). As well, similarity thresholds can be adjusted to user specifications for

enhanced viewing, and there is also the capacity to edit or alter “features” (such as exons). There are only a few disadvantages with this system: direction of exons and other features are not immediately interpretable, but this could be amended; Alfresco cannot search non-local databases (all current analyses access local databases); and this program does not support multiple pair-wise alignments (not yet possible using DBA).

### **Intronerator (displays WABA alignments, along with others)**

The Intronerator is a set of web-based tools, developed by Kent and Zahler (2000b), to supplement their WABA alignment program, by storing their *C. elegans-C. briggsae* alignments on this server. The Intronerator is an excellent tool for the nematode community. In the main display (Tracks Display – Figure 6 in appendix), users view (with useful zooming and scrolling options) the *C. elegans-C. briggsae* alignments, gene predictions from other sources like the Sanger AceDb (A *C. Elegans* DataBase), cDNA and EST alignments; these are viewed with the *C. elegans* genome as the reference. This server also has links to literature on various *C. elegans* genomic regions, allows retrieval of specific regions of the genome, and offers a small number of other databases and tools. Another useful feature (unique amongst the display tools) is the ability to align a nucleotide sequence of interest against *C. elegans* using WABA. The Intronerator was created to explore RNA splicing and gene structure in *C. elegans*, this species-specificity (resembling the Enteric/Menteric/Maj suite’s limitation) makes it an impractical tool for viewing comparisons to any other genome. The authors suggest that future work will direct them to allow gene discovery/display for other eukaryotic genomes, such as *Drosophila*, human and mouse.

### **VISTA – VISualization Tool for Alignment (displays GLASS alignments)**

VISTA was developed to display a multiple alignment in comparative studies of large (200 kb or more) genomic sequences from human, mouse and dog (Dubchak *et al.* 2000, Mayor *et al.* 2000 – Figure 7 in appendix). The alignments were accomplished using the GLASS algorithm for multiple pair-wise alignments followed by processing with intersection/union analyses to statistically determine conserved regions in all three genomes using length and percent identity thresholds along a sliding window (similar to PIP, but a continuous curve).

The versatility of this program does not match Alfresco or Intronerator in terms of interaction (it is a static display) or information displayed, offering only the graphical representation of the alignment along with an annotation (for the reference sequence only) of exon/intron locations. The strong points lie in its ability to handle gaps in any of the sequences (unlike PIPs) and its ability to visualize megabases of multiple alignments on the same scale. Unlike Alfresco however, it does not have the potential yet to display rearrangements or non-colinear regions. New improvements currently available are the ability to display multiple alignments, the capability to add a transcription factor binding site database search, and the addition of a new pair-wise alignment program based on both GLASS and MUMmer (Bray *et al.*, manuscript in preparation).

### **SynPlot (displays Dialign alignments, and more recently GLASS)**

A very similar visualization display tool called SynPlot (Gottgens *et al.* 2001) was developed with the Dialign alignment algorithm in mind. Like VISTA, SynPlot allows

the display of multiple alignments and shows the gaps in each sequence, as well as the nature and positions of conserved regions (based on percent identity of a sliding window) for all sequences. SynPlot has the added functionality of being able to display the features (exons, introns, repeat elements and CpG islands) for each sequence (Figure 8 in appendix). Again like VISTA, this program is restricted to colinear loci, provides only a static display, and is not suitable for comparing draft sequence (unlike PipMaker).

### **ACT – Artemis Comparison Tool (displays parsed BLAST alignments)**

ACT is a different type of sequence comparison viewer, whose program is available at Sanger's website: <http://www.sanger.ac.uk/Software/ACT/>. Along with the two sequences (input as one of a variety of file types), processed outputs from the standard BLAST alignment programs (usually BLASTn and tBLASTx) are used for the comparison of one or more pair-wise alignments. These alignments are processed with MSPcrunch (Sonnhammer and Durbin, 1994 – available as a server and as a program on several sites, including <http://www.cgr.ki.se/cgr/groups/sonnhammer/MSPcrunch.html>), a post-BLAST processing program primarily concerned with the proper treatment of similarities along large DNA sequences. MSPcrunch evaluates the BLAST Maximal Segment Pairs by applying a set of filtering rules to remove redundant and biased composition matches while keeping the weak matches if they are consistent with a larger gapped alignment. This interactive viewer is similar in display to the very nice graphic by Lee *et al.* (1998), who used an undisclosed, unavailable program. Another, less informative but similar visualization was published by Delcher *et al.* (1999b) using a different, but also unavailable program coupled to the MUMmer alignment program.

Although ACT can present several sequences, it can thus far only display pair-wise comparisons, unlike SynPlot and VISTA which process the pair-wise alignments into a multiple alignment before display. This is actually not a drawback: by allowing only pair-wise alignments, ACT is able to deal with the complex problems associated with displaying non-colinear sequences (unlike all the other viewers). Due to the prevalence of rearrangements even between two strains of the same species, this makes ACT indispensable for looking at whole genomes (Figure 9 and 10 in appendix). ACT is based on another DNA sequence viewer, an annotation tool called Artemis. With the added comparison(s), ACT is thus ideally suited to edit features of genomes (a secondary, annotation role).

Regions conserved in two genomes are “linked” by red homology blocks which are shaded depending on the similarity score (the user can set a display threshold). These blocks make interpretation of the comparison very clear, highlighting insertions/deletions and showing where rearrangements have occurred. The user can easily navigate by zooming and scrolling, and can center and align the two conserved regions by clicking on these red links. Inverted regions can also be flipped to make the regions colinear for easier comparison. Gene predictions and other genome features can also be displayed (in all six frames even). These features may be exported and used as queries for other analysis tools. For the above reasons, this particular tool was found to be the most attractive. ACT does not however, readily display draft sequence comparisons.

## Conclusion

There are now several different alignment algorithms available to compare two or more large nucleotide sequences of “genome-length”. Although all these programs take innovative approaches to find seeds in some way, from which they then extend their alignments, some algorithms solve the problem of large gaps and are more tailored for exon identification (WABA), others try to optimize speed or space utilization (most of the ones presented), some are best for highly conserved small regions (DBA, GLASS) or for large regions with only mild similarities (Dialign, LSH-ALL-PAIRS), yet others excel at finding all the differences between the compared sequences (MUMmer). It may be best to use a few alignment algorithms depending on the purpose of the study, or combine some of the features of several of these programs, like the new alignment program AVID (Bray *et al.* - in preparation). A thorough comparison of these algorithms has not yet been performed but would be of great value. Much work remains in this field; upcoming efforts should focus on: alignment algorithms with a rigorous statistical basis, improving the ability to handle multiple sequences, and proper measures of alignment accuracy for evaluation.

There are also a few varieties of high-caliber visualization tools for displaying such data, and again, these applications are normally tailored to the research interests. Some are strictly display databases (Enteric/Menteric/Maj suite), which may allow user input and search functions (Intronerator); others are primarily focused on a reference sequence, aligning one or more sequences to it along with analysis reports and features (PipMaker, VISTA, SynPlot); some allow interaction such as scrolling or real-time database searching (Alfredo, ACT); and some (at least ACT) allow visualization of the toughest alignment feature to capture, non-colinear blocks. As with alignment algorithms, combining the various qualities of these viewing tools would be of great scientific value; another tremendous utility would be to alter the source code of these programs to allow input from many or all of the alignment programs. Long-term objectives in this area should focus on improving graphic displays in an interpretable fashion and add the full complement of interactive features with seamless real-time searches/analyses with multiple local and remote databases.

## References:

- Alm, R. *et al.* 1999. *Nature* 397:176-180.
- Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. 1997. *Nucleic Acids Res.* 25:3389-3402.
- Badger, J. and Olsen, G. 1999. *Mol. Biol. Evol.* 4:512-524.
- Batzoglou, S., Pachter, L., Mesirov, J., Berger, B. and Lander, E. 2000. *Genome Res.* 10:950-958.
- Borodovsky, M., McIninch, J., Koonin, E. Rudd K., Medigue, C. and Danchin, A. 1995. *Nucleic Acids Res.* 17:3554-3562.
- Brenner, S. 1999. *Trends Genet.* 15:132-133.
- Buhler, J. 2001. *Bioinformatics* 17:419-428.
- Burge, C. and Karlin, S. 1997. *J. Mol. Biol.* 268:78-94.
- Casari, G., Andrade, M., Bork, P., Boyle, J., Daruvar, A., Ouzounis, C., Schneider, R., Tamames, J., Valencia, A. and Sander, C. 1995. *Nature* 376:647-648.
- Dandekar, T., Huynen, M., Regula, J., Ueberle, B., Zimmermann, C., Andrade, M., Doerks, T., Sanchez-Pulido, L., Snel, B., Suyama, M., Yuan, Y., Herrmann, R. and Bork, P. 2000. *Nucleic Acids Res.* 28:3278-3288.
- Delcher, A., Harmon, D., Kasif, S., White, O. and Salzberg, S. 1999a. *Nucleic Acids Res.* 27:4636-4641.
- Delcher, A., Kasif, S., Fleischmann, R., Peterson, J., White, O. and Salzberg, S. 1999b. *Nucleic Acids Res.* 27:2369-2376.
- Devos, D. and Valencia, A. 2001. *Trends Genet.* 17:429-431.
- Dubchak, I., Brudno, M., Loots, B., Pachter, L., Mayor, C., Rubin, E. and Frazer, K. 2000. 10:1304-1306.
- Dumas, J. and Ninio, J. 1982. *Nucleic Acids Res.* 10:197-206.
- Duret, L., Gasteiger, E. and Perriere, G. 1996. *Comput. Appl. Biosci.* 12:507-510.
- Fleischmann, R. *et al.* 1995. *Science* 269:496-512.
- Florea, L., Riemer, C., Schwartz, S., Zhang, Z., Stojanovic, N., Miller, W. and McClelland, M. 2000. *Nucleic Acids Res.* 28:3486-3496.
- Galili, N., Baldwin, H., Lund, J., Reeves, R., Gong, W., Wang, Z., Roe, B., Emanuel, B., Nayak, S., Mickanin, C., Budarf, M. and Buck, C. 1997. *Genome Res.* 7:17-26.
- Galperin, M. and Koonin, E. 1998. *In Silico Biol.* 1:55-67.
- Gelfand, M., Koonin, E. and Mironov, A. 2000. *Nucleic Acids Res.* 28:695-705.
- Glaser, P. *et al.* 2001. *Science* 294:849-852.
- Gottgens, B., Gilbert, J., Barton, L., Grafham, D., Rogers, J., Bentley, D. and Green, A. 2001. *Genome Res.* 11:87-97.
- Hardison, R., Oeltjen, J. and Miller, W. 1997. *Genome Res.* 7:959-966.
- Hayashi, T. *et al.* 2001. *DNA Res.* 8:11-22.
- Jareborg, N., Birney, E. and Durbin, R. 1999. *Genome Res.* 9:815-824.
- Jareborg, N. and Durbin, R. 2000. *Genome Res.* 10:1148-1157.
- Kent, W. and Zahler, A. 2000a. *Genome Res.* 10:1115-1125.

Kent, W. and Zahler, A. 2000b. *Nucleic Acids Res.* 28:91-93.

Kyrpides, N. and Ouzounis, C. 1999. *Mol. Microbiol.* 32:886-887.

Lee, I., Westaway, D., Smit, A., Wang, K., Seto, J, Chen, L., Acharya, C., Ankener, M., Baskin, D., Cooper, C., Yao, H., Prusiner, S. and Hood, L. 1998. *Genome Res.* 8:1022-1037.

Lund, J., Chen, F., Hua, A., Roe, B., Budarf, M., Emanuel, B. and Reeves, R. 2000. *Genomics* 63:374-383.

Mayor, C., Brudno, M., Schwartz, J., Poliakov, A., Rubin, E., Frazer, K., Pachter, L. and Dubchak, I. 2000. 16:1046-1047.

McClelland, M., Florea, L., Sanderson, K., Clifton, S., Parkhill, J., Churcher, C., Dougan, G., Wilson, R. and Miller, W. 2000. *Nucleic Acids Res.* 28:4974-4986.

Miller, W. 2001 *Bioinformatics* 17:391-397.

Morgenstern, B., Frech, K., Dress, A. and Werner, T. 1998. *Bioinformatics* 14:290-294.

Morgenstern, B., 1999. *Bioinformatics* 15:211-218.

Needleman, S. and Wunsch, C. 1970. *J. Mol. Biol.* 48:443-453.

Ogata, H. *et al.* 2001. *Science* 293:2093-2098.

Oggioni, M. and Pozzi, G. 2001. *FEMS Microbiol. Lett.* 200:137-143.

Perna, N. *et al.* 2001. *Nature* 409:529-533.

Read, T. *et al.* 2000. *Nucleic Acids Res.* 28:1397-1406.

Salzberg S., Delcher, A., Kasif, S. and White, O. 1998. *Nucleic Acids Res.* 26:544-548.

Schwartz, S., Miller, W., Yang, C-M. and Hardison, R. 1991. *Nucleic Acids Res.* 19:4663-4667.

Schwartz, S., Zhang, Z., Frazer, K., Smit, A., Riemer, C., Bouck, J., Gibbs, R., Hardison, R. and Miller, W. 2000. 10:5777-586.

Smith, T. and Waterman, M. 1981. 97:723-728.

Sonnhammer, E. and Durbin, R. 1994. *Comput. Appl. Biosci.* 10:301-307.

Sonnhammer, E. and Durbin, R. 1995. *Gene* 167:GC1-10.

Stojanovic, N., Florea, L., Riemer, C., Gumucio, D., Slightom, J., Goodman, M., Miller, W. and Hardison, R. 1999. *Nucleic Acids Res.* 27:3899-3910.

Tagle, D., Koop, B., Goodman, M., Slightom, J., Hess, D. and Jones, R. 1988. *J. Mol. Biol.* 203:439-455.

Uberbacher, E. and Mural, R. 1991. *Proc. Natl. Acad. Sci.* 88:11261-11265.

Vincens, P., Buffat, L., Andre, C., Chevrolat J.-P., Boisvieux, J.-F. and Hazout, S. 1998. *Bioinformatics* 14:715-725.

Xu, Y., Mural, R., Shah, M. and Uberbacher, E. 1994. *Genet. Eng.* 16:241-253.

Zhang, Z., Schwartz, S., Wagner, L. and Miller, W. 2000. *J. Comput. Biol.* 7:203-214.

## APPENDIX

### Availability of Alignment Algorithms

**ASSIRC** (Vincens *et al.* 1998)

Program only: <ftp://ftp.biologie.ens.fr/pub/molbio/>

**DIALIGN** (Morgenstern *et al.* 1998; Morgenstern 1999)

Program: <http://www.gsf.de/biodv/dialign.html>

Server: <http://bibiserv.TechFak.Uni-Bielefeld.DE/dialign/> or

<http://genomatix.gsf.de/cgi-bin/dialign/dialign.pl> or

<http://bioweb.pasteur.fr/seqanal/interfaces/dialign2-simple.html>

**DBA** (Jareborg *et al.* 1999)

Program: <http://www.sanger.ac.uk/Software/Wise2/dba.shtml>

Server: <http://www.sanger.ac.uk/Software/Wise2/dbaform.shtml>

**MUMmer** (Delcher *et al.* 1999)

Program only: <http://www.tigr.org/softlab/>

**PipMaker/BlastZ** (Schwartz *et al.* 2000)

Server only: <http://bio.cse.psu.edu/pipmaker/>

**GLASS** (Batzoglou *et al.* 2000)

Program and server: <http://plover.lcs.mit.edu/>

**WABA** (Kent and Zahler, 2000a)

Program and server: <http://www.soe.ucsc.edu/~kent/xenoAli/> or

<http://www.cse.ucsc.edu/~kent/xenoAli/>

**LSH-ALL-PAIRS** (Buhler 2001)

Not available on the Internet, must contact the author at: [jbuhler@cs.washington.edu](mailto:jbuhler@cs.washington.edu)

## **Availability of Comparative Alignment Viewers**

**PipMaker/BlastZ** (Schwartz *et al.* 2000)

Server only: <http://bio.cse.psu.edu/pipmaker/>

**Enteric/Menteric/Maj** (Florea *et al.* 2000, McClelland *et al.* 2000)

Server only: <http://glovin.cse.psu.edu/enterix/>

**Alfresco** (Jareborg and Durbin 2000)

Program and server: <http://www.sanger.ac.uk/Software/Alfresco/>

**Intronerator** (Kent and Zahler 2000b)

Server only: <http://www.cse.ucsc.edu/~kent/intronerator/>

**VISTA** (Dubchak *et al.* 2000, Mayor *et al.* 2000)

Program and server: <http://www-gsd.lbl.gov/vista/>

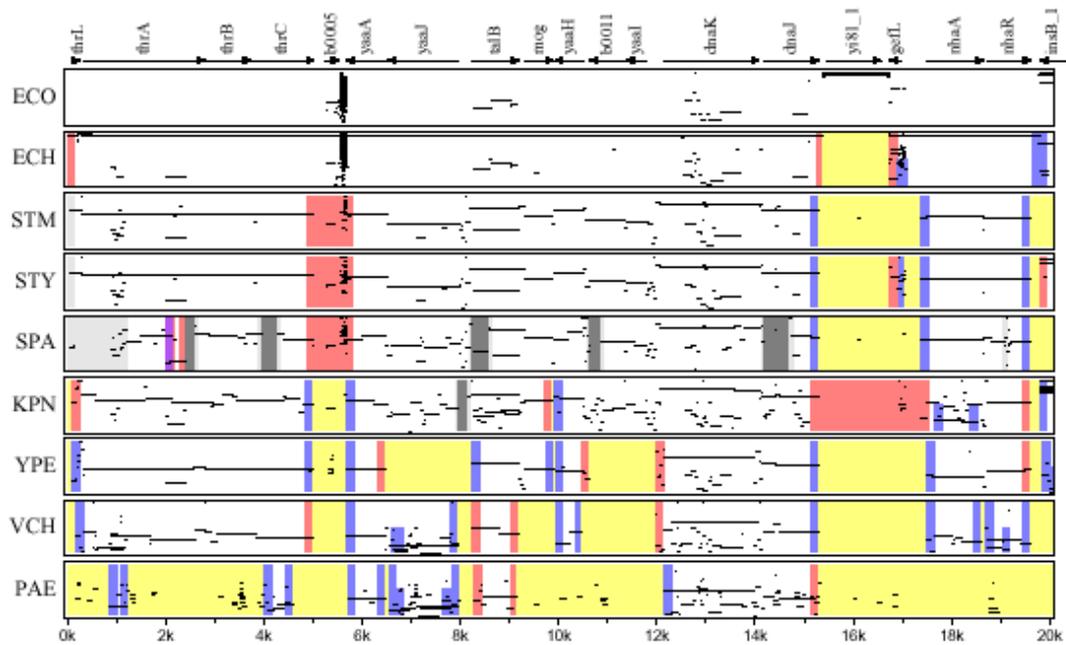
**SynPlot** (Gottgens *et al.* 2001)

Program only: <http://www.sanger.ac.uk/Users/jgrg/SynPlot/>

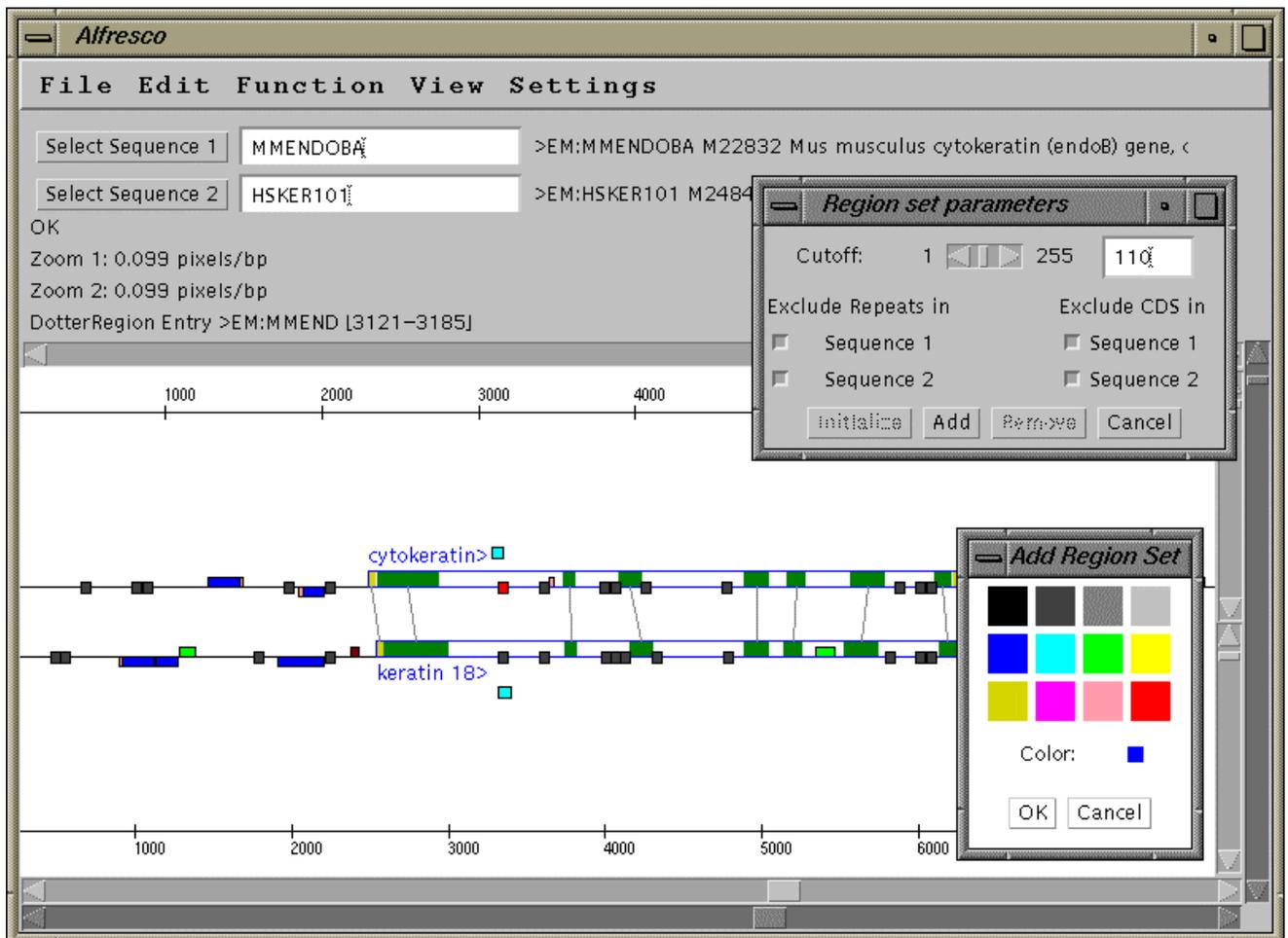
**ACT**

Program only: <http://www.sanger.ac.uk/Software/ACT/>

- yellow    E. coli sequence not found in the other species
- red        sequence in the other species whose immediate neighbor has a homolog elsewhere in E. coli
- blue      sequence in the other species whose immediate neighbor has no detectable homolog in E. coli
- gray      apparently not sequenced in the other species
- purple    overlapping colors, such as red and blue

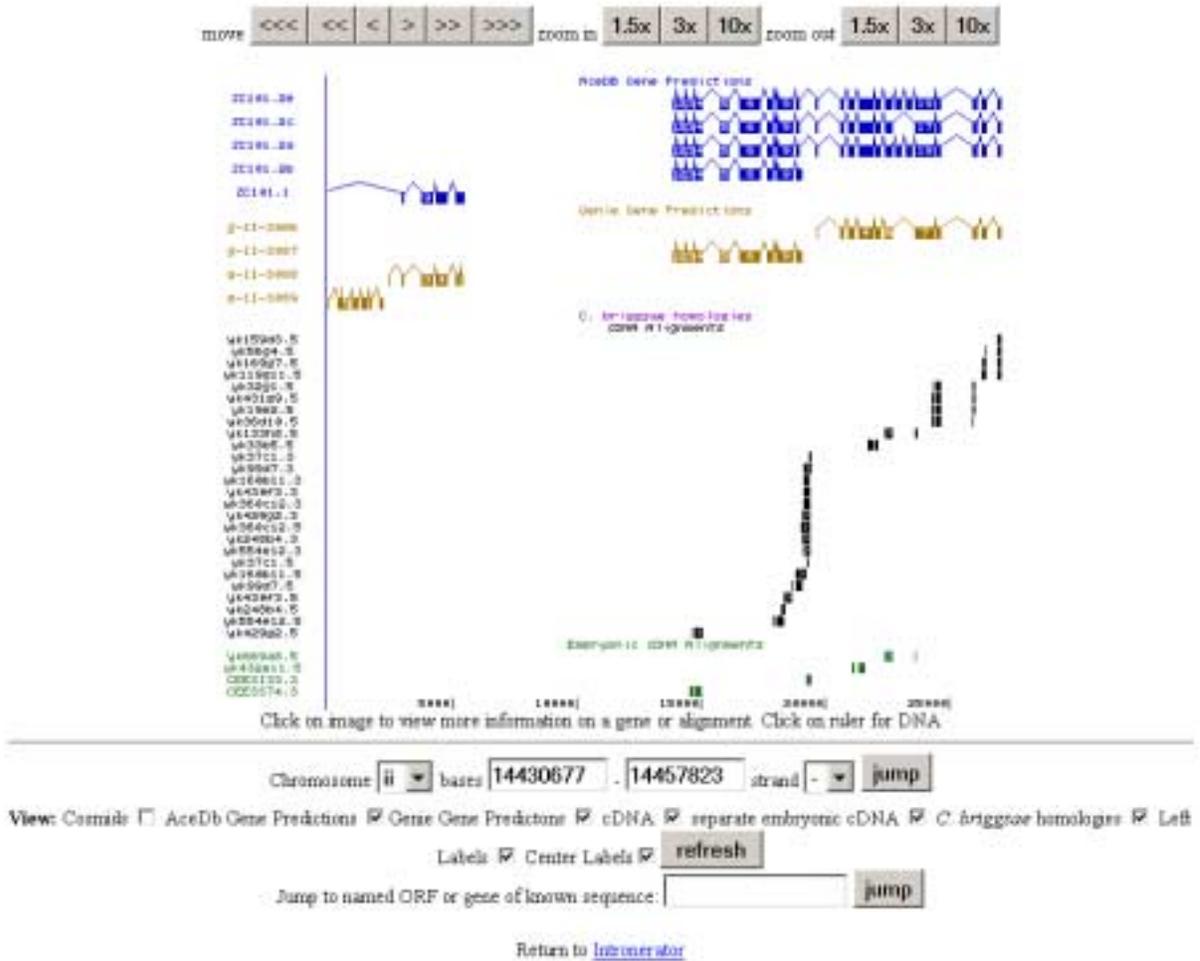


**Figure 4:** Enteric display of *E. coli* K12 (ECO) as the reference (centered on genome address 1111) and PIPs representing alignments to several related organisms below. Color legend and time of search are displayed at the top of the page, while the location of the genes in *E. coli* K-12 are displayed at the top of the graphic. ECH, *E. coli* O157:H7, STM, *Salmonella typhimurium* LT2, STY, *S. typhi*, SPA, *S. paratyphi* A, KPN, *Klebsiella pneumoniae*, YPE, *Yersinia pestis*, VCH, *Vibrio cholerae*, PAE, *Pseudomonas aeruginosa*.

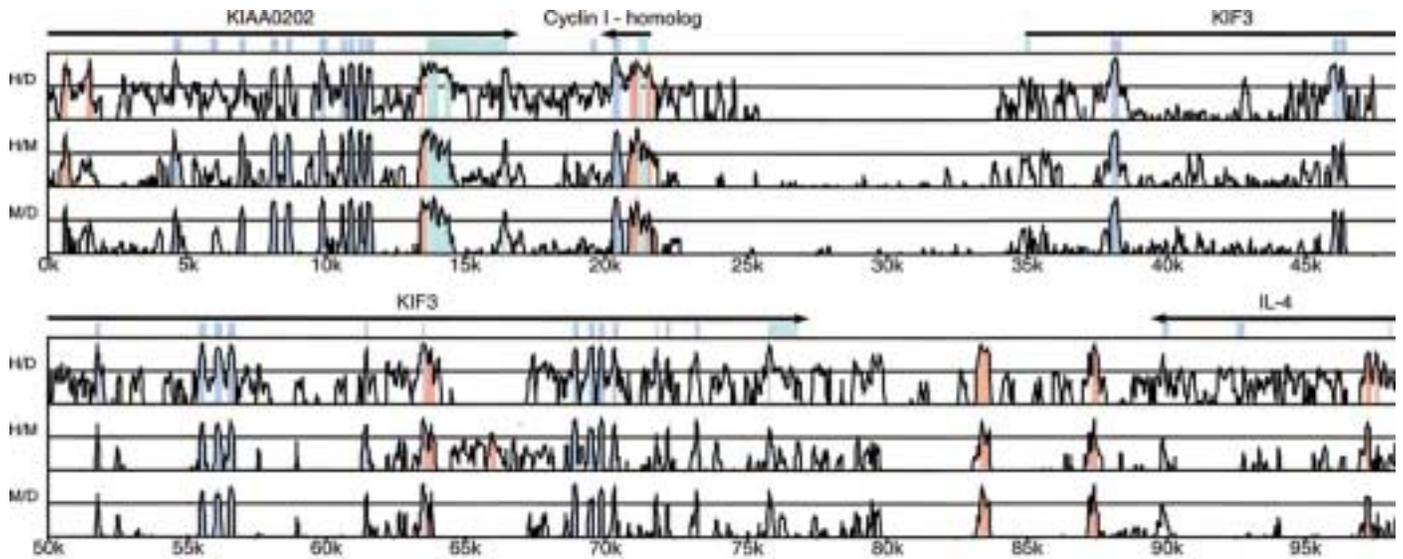


**Figure 5:** The Alfresco user interface. Similar regions can be identified using the “Regions set parameters” box. The cutoff scrollbar selects the threshold of similarity. The main display shows two EMBL sequence entries of orthologous mouse and human keratin 18 genes. Conserved regions are connected by gray lines. Boxes shaded a variety of blue represent conserved regions found by DBA. Sequence repeats are also indicated. (adapted from the web figure <http://www.sanger.ac.uk/Software/Alfresco/gfx/ismb4.gif>)

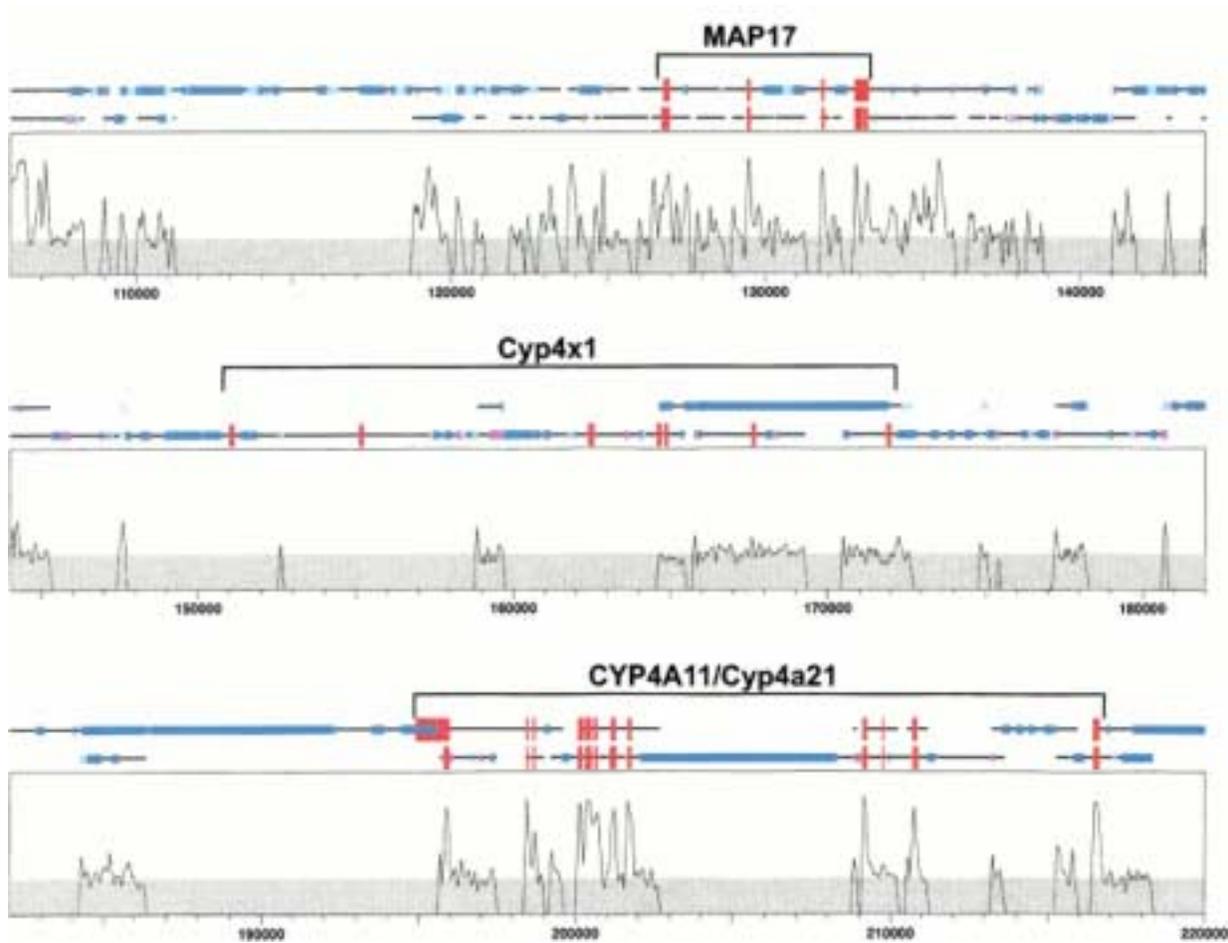
### Region Near ZC101



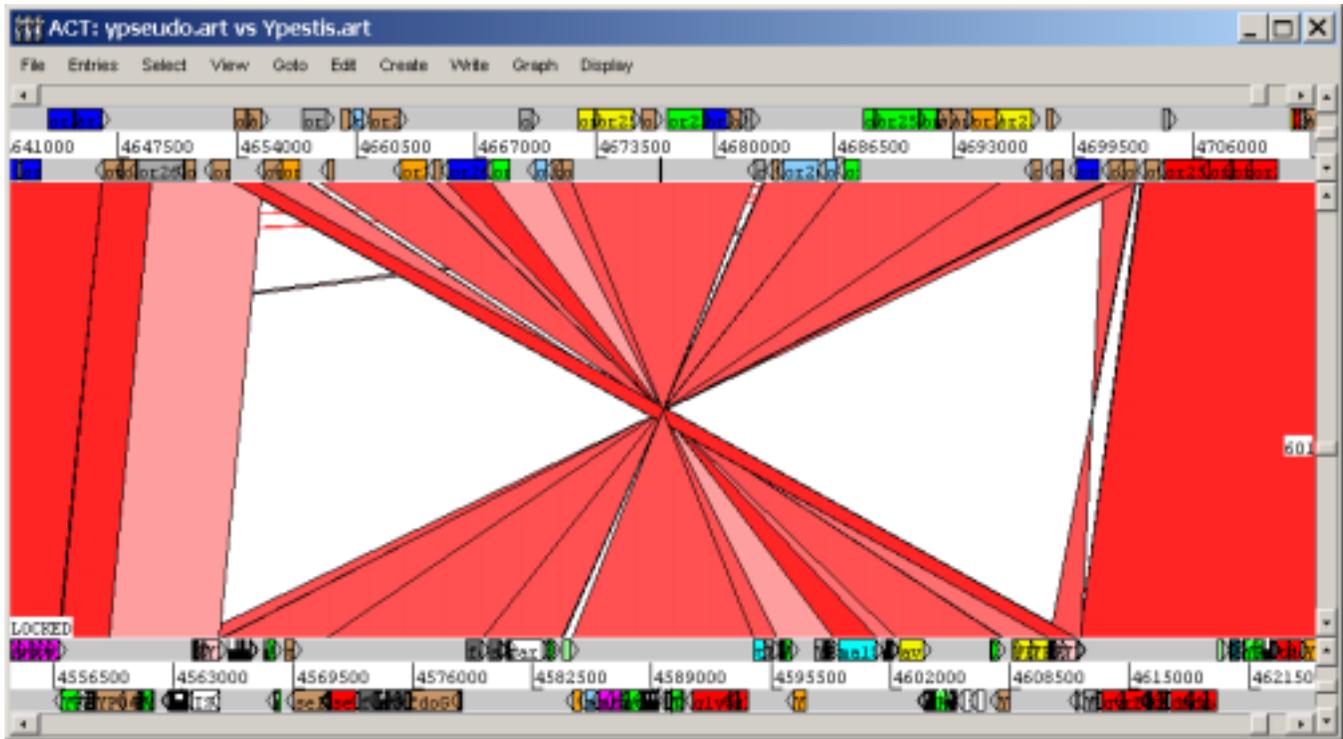
**Figure 6:** The Intronerator “Tracks Display” window, showing the region of the *C. elegans* genome associated with the entire cosmid ZC101. Buttons at the top of the page allow quick and easy scrolling and zooming. Below are displayed various gene predictions (based on AceDb, Genie) and homologies to *C. briggsae* (if any) as determined by WABA, and homologies to various cDNAs.



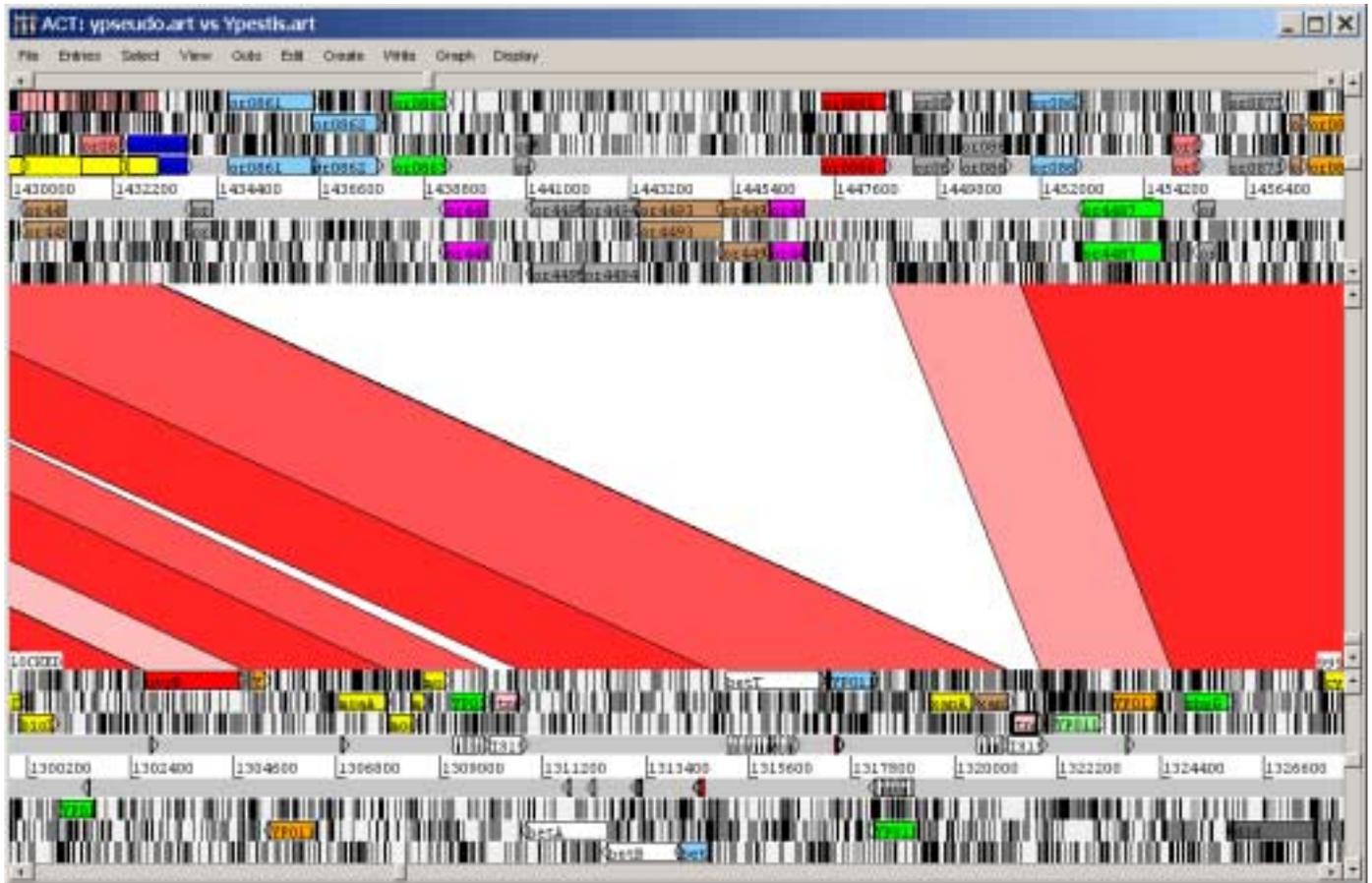
**Figure 7:** VISTA plots demonstrate peaks of similarity in the pair-wise sequence alignments, here between human and dog, human and mouse, and mouse and dog. Conserved sequences are shown relative to their positions in the human genome (the M/D alignment is mapped on the coordinates of the human genome sequence based on matching base pairs in the H/M alignment). Percent identities from 50%-100% are indicated along the vertical axes. Coding exons (blue rectangles) and 5' and 3' untranslated regions (turquoise) are shown above the profile. Peaks representing non-coding (red) and untranslated (turquoise) sequences fitting the criteria for conserved elements (as well as the blue coding sequences) are shown. (adapted from Dubchak *et al.* 2000)



**Figure 8:** SynPlot analysis of the human and mouse *CYP7A11/Cyp4a21* loci, as aligned by Dialign. Numbers on the vertical axis represent the proportion of identical nucleotides within a 49 bp window, moved in 25 nt increments across the entire alignment. The horizontal lines above the profile represent the human and mouse sequences and illustrate the position of gaps introduced to permit optimum alignment. Red boxes show exon positions, and the smaller boxes represent repeats. (adapted from Gottens *et al.* 2001)



**Figure 9:** ACT display of a *Y. pestis* vs. *Y. pseudotuberculosis* alignment. This figure clearly demonstrates an inversion of a 45 kb region, flanked by two co-linear segments. The different shades of red indicate the level of homology. Colored bars on the top and bottom indicate predicted genes in either species.



**Figure 10:** ACT display with the six reading frames shown. Colored boxes at the top and bottom again represent predicted genes in either species. This particular window displays what appears to have been a deletion event in *Y. pestis* with an insertion sequence remnant. One can postulate that this 15 kb region was once flanked by two of these IS elements (in parallel), such that recombination between the two would result in loss of the intervening region.