An applications-focused review of comparative genomics tools: Capabilities, limitations and future challenges

Patrick Chain, Stefan Kurtz, Enno Ohlebusch and Tom Slezak

Date received (in revised form): 24th February 2003

Patrick Chain
is responsible for the Biodefense Division's sequencing and Comparative Analysis Group within the Biology and Biotechnology Research Program at the Lawrence Livermore National Laboratory, Livermore, California.

Stefan Kurtz
is Professor for Computer Science in the Center for Bioinformatics at the University of Hamburg, Germany.

Enno Ohlebusch
is Professor for Theoretical Bioinformatics at the University of Ulm, Germany.

Tom Slezak
leads a Pathogen Bioinformatics team at the Lawrence Livermore National Laboratory.

Keywords: *comparative genomics algorithms, multi-sequence alignment, DNA signatures, microbial genome analysis, multi-genome alignment*

Abstract

A team at the Lawrence Livermore National Laboratory (LLNL) was given the task of using computational tools to speed up the development of DNA diagnostics for pathogen detection. This work will be described in another paper in this issue (see pages 133–149). To achieve this goal it was necessary to understand the merits and limitations of the various available comparative genomics tools. A review of some recent tools for multisequence/genome alignment and substring comparison is presented, within the general framework of applicability to a large-scale application. We note that genome alignments are important for many things, only one of which is pathogen detection. Understanding gene function, gene regulation, gene networks, phylogenetic studies and other aspects of evolution all depend on accurate nucleic acid and protein sequence alignment. Selecting appropriate tools can make a large difference in the quality of results obtained and the effort required.

A SURVEY OF GENOMIC ALIGNMENT AND COMPARISON TOOLS FOR WHOLE-GENOME PATHOGEN DNA SIGNATURE DEVELOPMENT

The output of sequence data from worldwide sequencing centres with constantly increasing sequencing capacities has been rising at an exponential rate for the past decade or two.[1] The first two publications of microbial whole genome sequencing projects appeared in 1995. Only eight years later, there are almost 100 completed bacterial genomes available.[2] Most of these are eubacteria and archaebacteria but there is also a completed yeast, along with draft analyses of several multicellular eukaryotes including nematode, fly, human, weed, mouse, mosquito, rat, rice, fungi and fish. More still are currently underway; the TIGR Microbial Database In-Progress[3] currently has over 180 entries and over 1,200 viral genomes have been submitted to NCBI.[4]

The increase in sequencing efficiency suggests that the bottleneck is not the accumulation of raw data but the annotation and analysis of sequences and genomes. There have been waits of longer than a year after the end of sequencing for some teams to analyse and publish genomes before the entire sequences were made publicly available. At least some of this delay can be attributed to the tools being used to perform the analyses. While the desire to postpone data release until publication is understandable, in some cases lengthy delays have prevented us from being able to create effective pathogen diagnostics in a timely fashion.

Given the continuing improvements in high-throughput genomic sequencing and

Tom Slezak,
Lawrence Livermore National Laboratory,
700 East Avenue L-448,
Livermore, CA 94550, USA

Tel: +1 925 422 5746
Fax: +1 925 422 2133
E-mail: slezak@llnl.gov

Accurate whole-genome comparisons are needed to solve many problems related to modern genomics

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. As noted above, these tools are also required for the specific problem of determining unique DNA and protein sequence for diagnostic assays. However, before any such detailed analyses can be addressed, methods are required for comparing and visualising such large sequences. These two topics are reviewed below.

COMPARATIVE GENOMICS

Every *alignment* is based on a collinear arrangement of sequence similarities

Alignments have been used to compare coding and non-coding regions from different species

A *genome comparison* does not assume colinearity of sequence similarities

A brief comment on terminology is in order. In our opinion, every *alignment* is based on a colinear arrangement of sequence similarities (Figure 1). All current alignment tools assume this colinearity. We will use a more general notion of *genome comparison* when we are discussing situations where an assumption of colinear similarities is not required (Figure 2). Searching for apparently unique regions of pathogen nucleic acid or protein sequence to create diagnostics is one such situation. To date the authors are aware of no general, versatile genome comparison tools; *ad hoc* techniques are used to identify syntenic regions, which then can be individually aligned. It is an overloading of terminology that 'comparative genomics' is a commonly used term that encompasses the capabilities of both *alignments* and *genome comparison* tools.

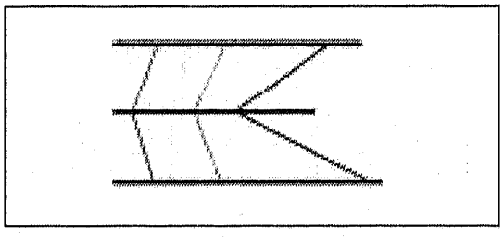

Figure 1: A cartoon representation of a set of colinear segments between three genomes

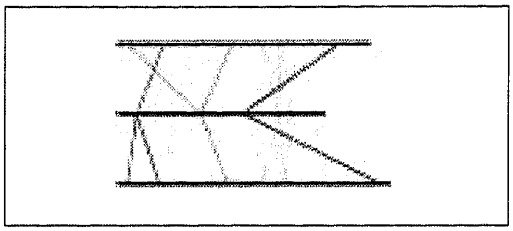

Figure 2: A representation of non-colinear relationships between homologous segments of three genomes

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 organisation of the genome and possibly have other, as-yet unknown, functions. Several comparative sequence analysis approaches using alignments have recently been used to analyse corresponding coding and non-coding regions from different species, although mainly between human and mouse.[5–14] 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. The use of alignment-based comparisons to uncover conserved functional elements has been termed 'phylogenetic footprinting'.[15] Of importance to annotation, this approach obviates the need for *a priori* knowledge of a protein sequence motif and provides a complement for algorithmic analyses.

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. This may prove especially important for exploiting common mechanisms of virulence, antibiotic

resistance and host range determination among pathogens. 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 incidents in the USA.[16] These forensic-level DNA signatures may employ single-nucleotide polymorphisms (SNPs), multi-locus variable-number tandem repeat analysis (MLVA) and similar variations. Note that this level of assay, which can sometimes track down to a single individual as the source of a fast-mutating strain/isolate, is generally inappropriate for the type of testing that attempts to answer the general question of: 'Does this air/water/food/blood sample contain any pathogens that we need to worry about?' Highly parallel detection using microarrays or equivalent techniques are now being used for precise strain determination.[17] The array of different research problems/goals highlights the need and utility of a versatile comparative genomics program/tool.

Alignment and substring matching programs are used to design DNA signatures for detection and forensic use

Comparison with near-neighbours genomes highlights important differences

Although it was once the goal to characterise 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 several publications of bacterial inter- and intra-species whole genome comparisons.[18–25] As an aside, we have found[26] that classical phylogenetics is an unreliable predictor of the ability or inability to construct DNA signatures that will detect all members of a family of species, or all strains of a species.

Aligning whole genomes requires a balance of speed and sensitivity

Underlying these genomic comparisons are alignment and substring matching programs, some of which have been recently developed to tackle the various problems of dealing with long nucleotide strings such as complete 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 visualisation/interpretation often 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 dedicated research and now 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[27] in 1970 and Smith–Waterman[28] in 1981 accomplished this using a dynamic programming approach, as respective examples of early global and local alignment algorithms.

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 and consume excessive space and/or time, though several companies offer specialised hardware to speed up Smith–Waterman and other algorithms.[29,30]

There is a typical trade-off between higher speed and increased sensitivity. Genome-length 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.

Whole genome comparisons must cope with rearrangements, large insertions/ deletions, and repeats

Some of the main problems associated with these goals lie in dealing with rearrangements (eg exon shuffling or other non-syntenous regions resulting from intramolecular recombinations), large insertions or deletions (sequences that share several regions of local similarity separated by unrelated regions), repeated elements (eg duplicated genes/ operons, transposons, SINES, LINES, etc), tandem repeats, and inherent problems of gene regulatory elements, including their small size and relative resistance to small insertions/deletions or substitutions. Another subject infrequently addressed for long sequences, and needing more in-depth exploration, is the issue of multiple alignments.

Few tools can align multiple whole genomes

Some or all of the above-mentioned problems are addressed by a number of recent programs discussed further below, such as DIALIGN,[31] ASSIRC,[32] MUMmer,[33] PipMaker/BlastZ,[34] GLASS,[7] WABA,[8] LSH-ALL-PAIRS,[35] Vmatch[36] and MGA.[37] Several of these programs are based on an efficient algorithm first used by Dumas and Ninio[38] in 1982, which finds exact matches of length k, called k-mers. Each k-mer is inspected to see if it is included in a longer match of some fixed minimum length, called a *seed*. The seeds are extended to form larger contiguous matches, possibly including substitutions and gaps.

Tools for visualising alignments are often geared to the particular research problems of the developers

Several comparative studies of genomes, or of large genomic segments, are still using older methodologies to solve their particular problem(s). For example, a recent study by Oggioni and Pozzi[39] opted to simply parse a BLASTn[40] search to identify three clone-specific blocks of sequence in a comparison of three *Streptococcus pneumoniae* serotypes. CrossMatch[41] was used by Lee *et al.*[42] 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.*,[6] a filtered dot-plot algorithm, using the program lineplot in the CGAT package,[43] 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 widespread exposure or acceptance by the general scientific community.

VISUALISING 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. Some tools also provide options to deliver the output in XML format. This simplifies parsing of the output and interfacing the programs with other tools. Independent of the format, with the large data sets used in comparing genomes, these results are most often not intuitively interpretable. Visualisation 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 centred on two-dimensional representations called dot-plots (LAD and LAV,[44] Dotter[45]), but the focus has since shifted to more compact, linear representations.[46,47]

Similar to alignment algorithms, the direction in development of new display tools often follows the goals of the research in question. In addition to an interpretable alignment, visualisation 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 biologists in their analyses.

Visualisation tools must choose whether, and how, to deal with multiple issues

Interactive features are other useful options to consider, such as the viewing resolution (a static graphic *v.* the ability to zoom in/out) and real-time analysis capabilities (eg ability to search specific regions for homologies). Other problems include: (1) how to represent breaks in synteny (such as genome rearrangements) if at all; (2) will alignments from both strands be displayed; (3) can and how will multiple alignments be shown; (4) is only one sequence the reference for the alignment(s); and (5) how will contigs be displayed if a non-finished genome is used as one (or more) of the entries for the alignment?

Anchor-based global alignment of whole genomes only succeeds if the inputs have very few arrangements

Unique input formats hinder wider use of visualisation tools

A further problem lies with the input of data for the visualisation programs, since most of these were developed to work on only one specific file format. Gottgens *et al.*[48] also raised the issue of availability of such software, as some undisclosed programs that generated figures shown in some publications[33,42] are not available. Several of these issues are addressed by a number of recent developments in comparative genome alignment visualisation programs such as PipMaker,[49] Alfresco,[50] Intronerator,[51] VISTA,[9,52] SynPlot[48] and ACT.[53]

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 visualisation tools that try to make a seamless transition from raw data to interpretable comparisons. As discussed in a review of genomic DNA sequence comparisons,[54] there remains much room for improvement in terms of long-sequence or whole-genome alignment (or multiple alignment) algorithms, and in terms of formatted or processed graphical output that a user may be able to interpret and combine with other analyses.

RECENT PROGRESSION OF GENOME SEQUENCE ALIGNMENT PROGRAMS

Almost all available alignment programs before 1996 were developed to find target regions similar to a single 'small' sequence. As already mentioned, every alignment is based on a colinear arrangement of sequence similarities (eg the order of segments of similar sequence is preserved, although the distances between those segments may vary among the inputs). Therefore, a global alignment of whole genomes makes sense only if the species are closely related (or more precisely, if very few genome rearrangements have occurred). All global alignment programs described below rely on this assumption. Several of these programs use an anchor-based method, which is divided into three phases:

- Computation of all potential anchors.
- Computation of an optimal colinear sequence of non-overlapping potential anchors: these are the anchors that form the basis of the alignment.
- Closure of the gaps in between the anchors.

For diverged genomic sequences, a global alignment strategy is probably predestined to failure for having to align non-syntenic and unrelated regions in an end-to-end colinear approach. In this case, either local alignments are the strategy of choice or one must first identify syntenic regions, which then can be individually aligned. An overview of the software tools discussed below can be found in Table 1.

DIALIGN: DIagonal ALIGNment

The era of large-scale alignment algorithms began in 1996 with the versatile alignment program called DIALIGN,[55] capable of both pairwise and multiple alignments. One of the novelties of this program was the use of gap-free whole segments for comparison, instead

Table 1: Alignment and comparison tools

Tool	Purpose	Comparison	Seeds	Seed extension	Visualisation
DIALIGN	Alignments	Multiple	Gap-free fragments	–	HTML on server, SynPlot
ASSIRC	Finding local similarities	Pairwise	*k*-mers	Random walk	–
MUMmer	Alignments	Pairwise	MUMs	–	DisplayMUMs
PipMaker	Alignments	Pairwise	*k*-mers	Greedy	PIP server
GLASS	Alignments of orthologous regions for exon prediction	Pairwise	*k*-mers	12 bp left and right of seed	VISTA, SynPlot
WABA	Local alignments for gene prediction	Pairwise	Two 8-mers (ignoring wobble bases) in 1 kb region	Pairwise HMM	Intronerator
LSH-ALL-PAIRS	Ungapped local alignments	Pairwise	Gap-free fragments	–	–
MGA	Alignments	Multiple	MultiMEMs	–	HTML via mga2html
vmatch	Finding local similarities	Pairwise	MEMs	Greedy or bounded by fixed number of errors	–

of using single bases or residues. The alignments are thus composed of gap-free segment pairs of equal length that would form diagonals in a dot-matrix comparison. Such segment pairs are sometimes called 'fragment alignments' or 'fragments'. A quality score is assigned to every possible fragment based on the probability of its random occurrence and the program tries to find a colinear collection of non-overlapping fragments with maximum total score.

DIALIGN uses a greedy algorithm without back-tracking to perform multiple alignments

For pairwise alignments, a collection of fragments with the maximum sum of scores (overall optimal alignment) is found by a modification of the standard dynamic programming scheme. For multiple alignments, a greedy algorithm is used. In a first step, all possible pairwise alignments are constructed. Next, the fragments contained in these pairwise alignments are sorted according to their scores, and according to the degree of overlap with each other, and then integrated one-by-one into a growing multiple alignment – provided they are colinear with those fragments that were previously integrated. Non-colinear fragments are discarded. When no additional fragment can be incorporated, gaps are introduced into the sequences to properly arrange the selected segment pairs. However, these gaps are not penalised.

Space/time problems can result if the input is too long or too 'deep'

The use of complete segments of sequences in the comparison allows DIALIGN to locate small conserved regions that cannot be detected by the standard Smith–Waterman alignment program. Consequently, the program is able to identify functionally important regions even in large genomic sequences. This particular feature was found useful by Gottgens *et al.*[56] in a long-range comparison of the mouse and human SCL loci. In their study, a visualisation tool called SynPlot was designed to display the DIALIGN alignments. The DIALIGN web server returns the alignment as HTML.

One drawback of the greedy procedure in DIALIGN is that once a fragment is incorporated into the alignment, it is fixed and cannot be removed. This can lead to misalignments, especially if sequences contain repeats. Like any global alignment tool, 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. However, in our experience it has been a great tool for aligning shorter viral pathogen complete genomes and less-than-genome fragment sequence. The authors have also used it to align multiple same-gene fragment sequences from bacterial genomes (useful when investigators submit the same gene region from multiple strains/isolates). When a given input is too long or too 'deep', DIALIGN can sometimes exhaust

either memory or the user's patience in waiting for results. For example, attempting to align all six variola genomes in Genbank, each 180+ kbp long, did not complete after one week. Very recently, an anchored alignment method has been incorporated into DIALIGN,[57] to overcome these kinds of problems. However, since this option is not yet available on the DIALIGN server, it is not clear how this method performs in practice.

ASSIRC – Accelerated Search for SImilarity Regions in Chromosomes

ASSIRC is fast and works well in the absence of large insertions and deletions

Suffix trees are used to locate maximal unique matches (MUMs)

Vincens *et al.*[32] developed a tool, called ASSIRC, to find regions of similarity in pairwise genomic sequence alignments. ASSIRC invokes three steps. (1) Pairs of identical *k*-mers are identified using standard hashing methods. (2) All *k*-mers 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 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 visualisation tool associated with it.

MUMmer – Maximal Unique Match (mer)

Short gaps between anchors are aligned. Long gaps represent insertions/deletions or repeats

MUMmer is a pair-wise anchor-based alignment program

A need to compare closely related bacterial species (strains even) motivated the creation of this pairwise anchor-based alignment program, capable of detecting every difference between two microbial genomes.[33] 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 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 *v.* a complete one). It proceeds in the following three phases: (1) A maximal unique match (MUM) decomposition of the two genomes G_1 and G_2 is computed. A MUM is a sequence that occurs exactly once in genome G_1 and once in genome G_2, and is not contained in any longer such sequence. Using the suffix tree of $G_1\$G_2$, MUMs can be computed in $O(n)$ time and space, where $n = |G_1\$G_2|$ and \$ is a symbol occurring neither in G_1 nor in G_2. (2) The matches found in the MUM decomposition are sorted, and the longest possible set of MUMs that occurs in the same order in both genomes is extracted, yielding the anchors. If there are m MUMs, then this can be done in $O(m \log m)$ time by an algorithm that finds the heaviest increasing subsequence (HIS) of a sequence of weighted integers. Note, however, that MUMmer actually uses a simpler $O(m^2)$ time dynamic programming algorithm. (3) The gaps in the anchor alignment that have length less than or equal to a given limit (5,000 bp is the default in MUMmer) are closed with a standard dynamic programming algorithm. For one gap consisting of two sequences of length r and r', this takes $O(rr')$ time and $O(\min(r,r'))$ space. Gaps that are longer than the limit remain unaligned. These consist of apparent insertions/deletions (lateral transfer, transpositions), polymorphic regions 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.

In our opinion, MUMmer was, and continues to be, a major breakthrough towards the solution of the alignment of two sufficiently similar genomic

The restriction of using only MUMs as anchors appears overly stringent

sequences. Like most other existing methods, however, MUMmer cannot align more than two genomic sequences. There are two other minor drawbacks. First, the restriction of using only MUMs as anchors seems unnecessary, and is not justified, since exact matches occurring more than once in a genome may also be meaningful. In fact, the coverage of the sequences increases if other matches are taken into account. Second, the use of the HIS algorithm for chaining the MUMs is not adequate. This is because the resulting chain of MUMs may contain overlapping MUMs, which in turn may lead to inconsistencies (ie it may not be possible to find an alignment that is consistent with all selected MUMs). MUMmer takes an *ad hoc* approach to handle this: it simply removes the overlapping parts from the MUMs.[37]

PipMaker can now pairwise compare multiple inputs to a single reference sequence

MUMmer2 has reduced space requirements, and companion program NUCmer can compare two draft genomes

Recently, a new version of MUMmer, MUMmer2, was developed.[58] While still limited to pairwise alignments, it improves on MUMmer by only computing the suffix tree of one sequence. This reduces the space requirement by about 50 per cent. The other sequence is matched against the suffix tree delivering maximal matches that are unique in one sequence, but may not be unique in the other sequence. MUMmer2 is currently unique in that it can compare an incomplete (draft) genome to either a finished genome or another incomplete genome using the companion NUCmer utility. Yet another program, PROmer, can align genomes whose proteins are similar but whose DNA is too diverse to align. This is accomplished by using a six-frame translation. A graphical interface, DisplayMUMs, has also recently been made available.

PipMaker – Percentage Identity Plot MAKER

PipMaker is only available via a server

A web server named PipMaker[34] 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 percentage identity plot or PIP. The underlying high-performance local alignment program Blastz[59] is a variant of the Gapped BLAST program[40] specifically designed for aligning two long sequences.

As previously mentioned, repetitive elements often wreak havoc with 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 be made available in the future.[54] (These features are now highly desirable, as many microbial genomes are only being taken to a draft level for economic reasons. This is especially true in the field of pathogen DNA signature development, where the cost of one completed reference strain sequence and up to nine draft strain sequences might equal the cost of three or four completed strain sequences.) One of PipMaker's latest features is the ability to enter multiple sequences; however, these sequences are all to be compared to the reference sequence in pairwise 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.

GLASS – GLobal Alignment SyStem

GLASS[7] also falls into the category of anchor-based alignment tools. It consists of five steps, which deliver a partial alignment of the two input sequences, possibly leaving some regions unaligned. (1) All pairs of exact matching *k*-mers (ie

strings of length k) in the two input sequences S_1 and S_2 are searched for. Initially, $k = 20$. (2) For a given pair of matching k-mers (w_1, w_2), its score $s = s_l + s_r$ is determined as follows: a dynamic programming algorithm is applied to the 12 nucleotides to the left of w_1 and w_2, yielding score s_l, and to the right of w_1 and w_2, yielding score s_r. (3) The highest scoring sequence of k-mers that occur in the same order in both DNA sequences is computed by a dynamic programming algorithm. (4) All k-mer matches in the resulting sequence whose score s is below a given threshold are removed. Furthermore, inconsistent overlapping k-mers are also removed; (w_1, w_2) and (w_1', w_2') are inconsistent, if the overlap of w_1 and w_1' differs from the overlap of w_2 and w_2'. (5) The resulting k-mers serve as anchors in the alignment of S_1 and S_2. Steps (1)–(5) are recursively applied to the gaps (all unaligned regions between the anchors) with decreasing values of k, namely 15, 12, 9, 8, 7, 6, 5. Finally, any remaining gaps are aligned by standard dynamic programming techniques.

GLASS is anchor-based and appears to be limited by large space requirements

A major drawback of GLASS is the large space requirement. For example, an alignment of two DNA sequences of human and mouse (222,930 bp and 227,538 bp) is produced in about 14 minutes using 1.14 gigabytes of main memory. It takes 38 minutes to align a similar pair of sequences of twice the length, using 2.05 gigabytes. Furthermore, it seems that GLASS does not take advantage of long identical regions in sequences. For example, to deliver an alignment of the initial 200,000 bp of two strains of *Escherichia coli*, GLASS requires more than 25 hours of computation time (the job was stopped after 25 hours). Although no visualisation tool was presented with the program, subsequent research using GLASS did present the modified output with VISTA.[9] Here, the authors used GLASS in a three-way comparative study involving human, mouse and dog, but the alignments were done in pairwise fashion and intersection/union analyses were performed to find regions conserved in all three sequences.

WABA – Wobble Aware Bulk Aligner

WABA[8] is limited to pairwise alignments. The key feature of WABA is that wobble bases are treated differently from other bases. The third base in a codon is called *wobble base* because mutations in this base are often silent in the sense that they do not change the corresponding amino acid (due to the redundancy of the genetic code). WABA has been developed specifically for separately aligning 229 different sequences from *Caenorhabditis briggsae* (8 megabases total length, 34,722 bp average length) against 97 megabases of the *C. elegans* genome. These are two closely related nematodes.

WABA can be divided into three phases. (1) The smaller of the two input sequences is broken into short, overlapping sequence fragments. Then homologies between the short sequence fragments and the other input sequence are found. 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 that are scored for the best local alignment. (2) Homologous regions are aligned in an extended window using a pairwise hidden Markov model.[60] (3) If any two of these local alignments overlap by at least 15 bp and are identical in the overlapping region, then they are merged into one larger alignment.

WABA treats wobble bases differently and thus works well to uncover conserved exons

WABA is the first alignment tool that accounts for divergence in the wobble position of coding regions, and thus works well to uncover conserved exons. A visualisation tool called Intronerator

was also designed by the authors to aid in interpretation of the alignments.

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

Vmatch uses enhanced suffix arrays to efficiently solve large-scale string-matching problems on a pair of sequences

LSH-ALL-PAIRS[35] was designed specifically to 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 randomised 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.

LSH-ALL-PAIRS can find similar sequences using a long seed length while allowing several substitutions

This program is run iteratively to minimise 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.

Maximal exact matches (MEMs) are computed directly, providing a large speed improvement over hash/extend methods

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 visualisation tool has not yet been addressed for use with this program, and the program itself is not available for use or downloading.

Vmatch – index-based large-scale matching

Vmatch[36] is a new tool that solves a variety of large-scale string-matching problems on pairs of sequence sets very efficiently. The basic concept is to preprocess a set of database sequences into an enhanced suffix array, which provides a very powerful index structure for string matching. In a recent paper by Abouelhoda *et al.*[61] it is shown that several string-matching algorithms, originally developed for suffix trees, can be adapted to enhanced suffix arrays. The advantage of enhanced suffix arrays over suffix trees is a considerably reduced space requirement and a much faster processing in practice. Vmatch is accompanied by a tool, mkvtree, which computes an enhanced suffix array for a set of database sequences and stores this on file. Vmatch reads these files as a database and matches query sequences against the database sequences to find local similarities. To do this efficiently it utilises a new algorithm[36] to compute maximal exact matches (MEMs), ie exact matches between the database sequences and the query sequences that cannot be extended further without a mismatch. Unlike the hashing methods (which first generate *k*-mers and then extend these to MEMs) the new algorithm directly computes MEMs. As a consequence, when comparing large genomes or genome sets, Vmatch is much faster (by a factor between 6 and 24) than previous tools utilising the traditional hashing methods.

Vmatch also follows the find-seed-and-extend approach, with the seeds being MEMs. These are extended using the greedy algorithm of Zhang *et al.*[62] or alternatively by the method used in REPuter.[63]

A flexible range of options and the ability to handle very large sets of

sequences gives Vmatch an ability to efficiently compare a single draft or completed microbial genome against other microbial genomes. This is done by considering the non-target genomes as the database sequences and the target genome as a query sequence. Utilising an additional post-processing option of Vmatch, one can find all substrings of the target genome over a specified threshold length that have no match in any other genome. Other options allow a degree of non-exact matches also to be excluded from the output, in order to select those substrings in the target genome that do not even have any close match in the non-target genome. These unique capabilities of Vmatch were the key to automating the task of developing pathogen DNA diagnostics. Inputs ranging from raw sequencer reads to complete 5 Mbp bacterial genomes have been compared against concatenated genomes of greater than 800 Mbp using this technique.

Vmatch can compare a 5 mb pathogen genome against an 800+ mb concatenation of all other microbial genomes to find unique pathogen sequence regions

MGA – Multiple Genome Aligner

In contrast to virtually all known methods, MGA[37] can produce a global multiple alignment of whole genomes of closely related species. As several of the alignment programs described above, MGA uses an anchor-based method.

Enhanced suffix arrays are used to compute the anchors

MGA determines maximal exact matches (anchors) present in all input genomes in the same order

In the first phase of MGA, a novel algorithm detects all *maximal multiple exact matches* (*multiMEMs*) whose length exceeds a given threshold. In short, a multiMEM is a sequence of length l that occurs in all genomes $G_1, \ldots, G_k$ (at positions $p_1, \ldots, p_k$) and cannot simultaneously be extended to the left (left maximality) or to the right (right maximality) in every genome. A multiMEM will be denoted by a $(k+1)$-tuple $(l, p_1, \ldots, p_k)$. In principle, multiMEMs can be computed as follows: (1) construct the suffix tree of $S = G_1\$_1G_2\$_2 \ldots G_{k-1}\$_{k-1}G_k\$$, where $\$_1, \ldots, \$_{k-1}$ are pairwise different separator symbols not occurring in any of the genomes (this takes only $O(n)$ time and space, where $n = |S|$). (2) For every node α of the suffix tree, compute in a bottom-up fashion the set P_α of all positions in S where the sequence u corresponding to α starts. (3) Divide P_α into pairwise disjoint and possibly empty position sets $P_\alpha(q) = \{i \in P_\alpha | i$ is a position in $G_q\}$, where $1 \leq q \leq k$. If all position sets $P_\alpha(q)$ are non-empty, then every tuple $(|u|, p_1, \ldots, p_k)$ with $p_q \in P_\alpha(q)$ for all $q \in [1, k]$ denotes a multiple exact match because this implies that u occurs in every genome G_q at position p_q. Right maximality of this multiple exact match can easily be ensured during the incremental computation of the sets $P_\alpha(q)$. So the multiple exact match is a multiMEM if and only if it is left maximal. This is checked by comparing the characters immediately left to the positions p_q, $q \in [1, k]$. Thus, the algorithm takes time $O(kn + r)$ to compute all multiMEMs, where r is the number of right maximal multiple exact matches. In practice, however, it suffers from the huge space requirement of the suffix tree. Consequently, our efficient implementation is based not on suffix trees but on enhanced suffix arrays[61] that require only 5 bytes per input character.

In the second phase, MGA computes the 'anchors', consisting of the best non-overlapping sequence of multiMEMs that occur in the same order in each of the genomes as follows. Let $\{c_1, \ldots, c_m\}$ be the set of multiMEMs computed in the first phase. View every multiMEM $(l, p_1, \ldots, p_k)$ as a k-dimensional cube c in the Euclidean space with associated weight $w(c) = l$. A cube $c = (l, p_1, \ldots, p_k)$ precedes cube $c' = (l', p'_1, \ldots, p'_k)$ if and only if for all $p_i + l < p'_i$ for all $i \in [1, k]$. If c precedes c', denoted by $c \prec c'$, then c and c' are non-overlapping. A subset $C = \{c_{i_1}, \ldots, c_{i_q}\}$ of $\{c_1, \ldots, c_m\}$ satisfying $c_{i_1} \prec c_{i_2} \prec \ldots \prec c_{i_q}$ is called a *chain*, and the weight of C is $\sum_{j=1}^{q} w(c_{i_j})$. In order to find the best non-overlapping sequence of multiMEMs, one has to find a chain with maximum weight among all

chains. A well-known solution to this problem consists of constructing a weighted acyclic directed graph $G = (V, E)$ with vertices $V = \{\text{start}, c_1, \ldots, c_m, \text{stop}\}$.

The set of edges E is characterised as follows. There is an edge $start \rightarrow c_1$ with weight 0 for $j \in [1, m]$, an edge $c_i \rightarrow c_j$ with weight $w(c_i)$ if $c_i \prec c_j$ and an edge $c_i \rightarrow stop$ with weight $w(c_i)$ for $i \in [1, m]$. A maximum weight chain of cubes corresponds to a path with maximum weight from vertex *start* to vertex *stop* in the graph. Because the graph is acyclic and there are $O(m^2)$ edges, such a path can be computed in time $O(km^2)$, ie the running time of the algorithm is quadratic in the number m of multiMEMs. This can be a serious drawback if m is large. To overcome this obstacle, a variant of an algorithm devised by Zhang *et al.*[64] was recently implemented. This algorithm takes advantage of the geometric nature of the problem. It is based on *kd*-trees, a data structure known from computational geometry.[65] As is typical with *kd*-tree methods, no rigorous analysis of the running time of the algorithm is known. However, our experiments show that it gives a considerable increase in speed in practice.

MGA works well at aligning similar bacterial and large DNA viral genomes. Short RNA viral genomes often have mutation rates too high to permit conserved anchors

Gaps are shortened by recursive application with shorter anchor lengths

In the third phase, MGA closes the gaps between the anchors. First this is done by recursively applying the same method a certain number of times, thereby lowering the length threshold for the multiMEMs. The gaps that are left are handled as follows. Long gaps remain unaligned in order to cope with long insertions, deletions, etc. Short gaps are closed by a standard multiple sequence alignment program. The program ClustalW[66] was chosen for this task because it is a widely used implementation of profile-based progressive multiple alignment. It is easy, however, to interface other multiple alignment programs with MGA.

Short gaps between anchors are closed by ClustalW. Large gaps indicate long insertions/ deletions or repeat regions

Of course, MGA should also be supplemented with a good interactive visualisation component. It should be possible to adapt some of the visualisation techniques employed in the REPuter program[63] to obtain an alignment browsing system, which gives a good overview of an entire *pairwise* alignment of large sequences and also allows zooming in and out on regions of interest. The visualisation of multiple alignments of such a large size, however, remains a challenge. A current ability to render MGA outputs as HTML is available.

In practice, MGA has worked extremely well at aligning similar bacterial and (large) double-stranded DNA viral genomes. Aligning large gaps between anchors may lead to unacceptable running times. Many (short) single-stranded RNA viral genomes have such high mutation rates that MGA is unable to find enough anchors; in extreme cases, searching for very short anchors can exhaust memory. The requirement that all anchors must exist in all inputs is a drawback for DNA signature development instances where it would be highly desirable to align one or more finished genomes with a collection of sequence fragments (eg contigs from draft genomes or gene sequence fragments from Genbank).

THE PROGRESSION OF VISUALISATION TOOLS FOR DISPLAYING GENOMIC COMPARISONS

As mentioned above, 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. Such tailored tools are often not appropriate, or must be redesigned to suit new project goals. There remains a severe lack of versatile visualisation tools to serve the needs of the average molecular biologist with different research interests and display needs.

PIP (displays PipMaker's Blastz alignments)

A PIP is a representation of all the local alignments between two sequences and

their qualities (as measured by its percentage 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 (percentage 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 positions of the alignments in the other, non-reference sequences can be accessed in several ways, including pop-up messages if viewed in the Adobe Acrobat® reader with the proper PIP option requested. 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.

PIP has useful options, but a static display and limitations of server-based access are drawbacks

Alfresco (displays alignments from several sources)

Intronerator presents *C. elegans–C. briggsae* alignments

The goal in developing Alfresco[50] was to provide an interactive graphic front-end for a variety of analysis programs as they pertained to comparative analysis. In addition to a variety of displays (overview graphic, textual alignment and dot-plot), Alfresco can combine alignments with processed BLASTn results, BLASTx hits, repeats, results from gene modellers, expressed sequence tag hits and CpG islands. Selected regions can be subjected to further analysis. 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, which can be found on the Sanger Institute website.[67]

Alfresco is an interactive front-end to a variety of analysis programs

Conserved regions are shown by colour and are attached by a line, which suggests an ability to represent non-colinear regions. 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; and this program does not support multiple pairwise alignments.

Intronerator (displays WABA alignments, along with others)

The Intronerator is a set of web-based tools[51] developed by Kent and Zahler to supplement their WABA alignment program, by storing their *C. elegans–C. briggsae* alignments on this server. The Intronerator was created to explore RNA splicing and gene structure in *C. elegans* and is an excellent tool for the nematode community. In the main display, users view (with useful zooming and scrolling options) the *C. elegans–C. briggsae* alignments, gene predictions from other sources such as 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.

VISTA – VISualisation Tool for Alignment (displays GLASS alignments)

VISTA can now display multiple alignments

VISTA[52] was developed to display a multiple alignment in comparative studies of large (200 kb or more) genomic sequences from human, mouse and dog.[9] The alignments were accomplished using the GLASS algorithm for multiple pairwise alignments followed by processing with intersection/union

analyses to statistically determine conserved regions in all three genomes using length and percentage 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, since it is a static display 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 colinear sequences (unlike PIPs) and its ability to visualise 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 pairwise alignment program based on both GLASS and MUMmer.[68]

SynPlot can display multiple alignments with annotation features highlighted

SynPlot (displays DIALIGN alignments and, more recently, GLASS)

A very similar visualisation display tool called SynPlot[48] 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 percentage 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. Again like VISTA, this program is restricted to colinear segments, provides only a static display, and is not yet suitable for displaying draft sequence comparisons.

ACT supplements a pair-wise alignment display with parsed BLAST information and can deal with non-colinear inputs

ACT – Artemis Comparison Tool (displays parsed BLAST alignments)

ACT is a different type of sequence comparison viewer, whose program is available at the Sanger Institute website.[67] Along with the two sequences (input as one of a variety of file types), processed outputs from the standard BLAST alignment programs BLASTn or tBLASTx are used for the comparison of one or more pairwise alignments. These alignments are processed with MSPcrunch,[69] 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.*,[42] who used an undisclosed, unavailable program. Another less informative but similar visualisation was published by Delcher *et al.*[33] using a different, but then-unavailable, program (display MUMs) coupled to the MUMmer alignment program.

ACT can present several sequences as pairwise comparisons, unlike SynPlot and VISTA which process the pairwise alignments into a multiple alignment before display. This is not a drawback: by allowing only pairwise alignments, ACT is able to deal with the complex problems associated with displaying non-colinear sequences (unlike all the other viewers). Owing to the prevalence of rearrangements even between two strains of the same species, this makes ACT unique for looking at whole genome comparisons. ACT is based on, and maintains the same functionalities as another DNA sequence viewer, an annotation tool called Artemis.[70]

Regions conserved in two genomes are 'linked' by 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 has the option to centre and align two conserved regions by clicking on these links. Inverted regions can be flipped for easier, more interpretable comparisons. Gene predictions and other genome features can also be displayed (in all six frames even). All features may be exported and used as queries for other analysis tools.

Current alignment tools cannot exploit all available sequence (whole genomes, draft genomes, and fragments)

DisplayMUMs

This tool has recently been released by TIGR as version 1.0 research software.[71] Its primary uses are for examining sequence assembly (comparing a new assembly to an old one, for example) and for visualising polymorphisms between two aligned genomes.

We need an ability to align multiple sequences with relaxed restrictions of colinearity and presence of anchors in all inputs

Visualising MGA alignments

MGA has an option to output an alignment in XML format. This can be turned into HTML using the program mga2html. It is available, along with examples and instructions, on the MGA web site.[72]

FUTURE DIRECTIONS: OPEN ISSUES FOR PATHOGEN GENOME COMPARISON

Multiple sequence alignments must be scaled up to deal with large input sets

There are a number of tools and capabilities that are sorely needed to be able to achieve all our goals for reliable DNA and protein signature creation.

Aligning whole genomes with fragment sequences

As discussed above, we currently do not have alignment tools that can align all the whole genomes and gene-focused fragment sequences that are available for an ever-increasing number of organisms. Thus, we are forced to either use crude techniques or be unable to exploit all the information available to us. (MUMmer2 could align a set of fragment sequences against a single whole genome, which is a limited version of our need. Sets of gene fragment sequences are not equivalent to the draft genomes that MUMmer2 expects, which could be problematic.)

Comparing more distantly related genomes (dealing with non-colinearity)

The tools we rely on in our current pipeline are not capable of dealing with all the types of large-scale genome rearrangements that occur among distantly related (and even some closely related) genomes. In practice, this limits our ability in some cases to analyse highly divergent families to seek DNA diagnostics that can identify all members of that family. It also hinders our ability to use multiple sequence alignments to rapidly discover common mechanisms that may have mutated significantly yet still should be recognisable as having a common origin.

Finding commonality amongst apparently unrelated or distant genomes (data mining for shared mechanisms)

Our current tools require a rather high degree of commonality, distributed heavily across the entire genome span, to allow alignment. We would like to be able to enter hundreds or thousands of microbial genomes into a tool and let it discover all clusters of similar regions, separated by a large amount of evolutionary time.

Massive scaling of alignments

Within this decade it will become common to have dozens or even hundreds of completed genomes for viruses. Even today, current tools are not capable of aligning all the HIV genomes that are in Genbank. (The extreme divergence, recombination and lack of DNA repair mechanisms in HIV make it impossible for an anchor-based multiple alignment technique such as MGA to locate common anchors, and non-anchor alignment techniques such as DIALIGN choke on the more than 200 HIV genomes now available.) Space/time constraints need to be constantly pushed back as the cost of acquiring genomic sequence continues to plummet.

Comparison of multiple draft genomes

Efficient comparison of multiple draft genomes is needed

No tools are yet available to adequately compare multiple draft genomes, either among themselves or with completed genomes. (As noted above, Vmatch can find exact match substrings between such inputs, but this is only a special case of what would be desired. MUMmer2 can compare two genomes at any stage of completion, which is a promising start for our needs.) Large genome centres have already demonstrated the ability to draft sequence a bacterial genome in one day or less, and the cost of finishing a genome can drastically exceed the cost of the draft. Hence, we anticipate massive amounts of draft microbial sequence in the near future, with the hope that new tools will allow them to be correctly aligned against one or more completed reference sequences (same species or near-neighbour), or compared among themselves if no reference is available.

Better dealing with inexact matches

A large-scale and efficient electronic PCR capability is needed to determine when 'close enough for PCR' matches exist for apparently-unique sequence

Current tools such as Vmatch and MUMmer are excellent at finding exact substring matches. Vmatch can also find approximate matches, but it uses a simple unit score function that does not account for the specific constraints of our application. In practice we have found that current tools cannot adequately prevent us from selecting candidates that have enough differences from some near-neighbour sequence to pass our electronic screening, yet fail in practice in the wet laboratory. A complete solution to this would be an electronic polymerase chain reaction (PCR) routine that takes into account all the factors involved in hybridisation, yet this probably requires a supercomputer. A good heuristic program that properly predicts cross-hybridisation in at most a few hours, on affordable hardware, is one of our wishes.

Use of massive new computers

Considerable opportunities exist for practical genome comparison tool development

The Lawrence Livermore National Lab recently announced a procurement with IBM that will result in two of the world's most powerful computers.[73] One of these, Blue Gene/L, will have 130,000 central processing units (CPUs). To truly exploit this enormous computational power, new highly parallel multiple alignment algorithms must be developed. It remains a challenge to divide the multiple alignment task into subproblems which can each be solved on a single CPU, minimising communication with other CPUs. Such algorithms would permit tackling many interesting alignment problems that could not otherwise be solved in reasonable time.

CONCLUSIONS

A wide range of genome comparison tools is now available. One way to categorise them is:

- pairwise local alignment comparison tools;
- global alignment tools: pairwise alignment, multisequence alignment and multigenome alignment;
- substring maximum-exact-match tools;
- alignment viewing tools.

It has been noted that all of these tools have their own niches, advantages and limitations. A practical use of several of these tools to aid in the design of DNA diagnostics for pathogens has been described in a separate paper in this issue. In our application, MGA is currently used to align multiple large, similar genomes; DIALIGN is used to align viral genomes where adequate anchors cannot be determined; and Vmatch is used to determine which portions of the alignments determined by MGA and DIALIGN are apparently unique when compared against all other available microbial genomes. A number of areas are listed in which specific improvements or new capabilities are needed to meet the challenges presented by the onslaught of new sequence data.

Acknowledgments
We thank Mohamed Ibrahim Abouelhoda for his helpful comments on earlier versions of the paper. Thanks also to Burkhard Morgenstern for his suggestions improving the description of DIALIGN. This work was performed under the auspices of the US Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

References

1. Genbank Statistics (URL: http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html).
2. Bacterial Genomes (URL: ftp://ftp.ncbi.nih.gov/genbank/genomes/Bacteria/).
3. TIGR Genomes in Progress (URL: http://www.tigr.org/tdb/mdb/mdbinprogress.html).
4. NCBI Viral Genomes (URL: http://www.ncbi.nlm.nih.gov/PMGifs/Genomes/vis.html).
5. Hardison, R., Oeltjen, J. and Miller, W. (1997), 'Long human–mouse sequence alignments reveal novel regulatory elements: A reason to sequence the mouse genome', *Genome Res.*, Vol. 7, pp. 959–966.
6. Lund, J., Chen, F., Hua, A. *et al.* (2000), 'Comparative sequence analysis of 634 kb of the mouse chromosome 16 region of conserved synteny with the human velocardiofacial syndrome region on chromosome 22q11.2', *Genomics*, Vol. 63, pp. 374–383.
7. Batzoglou, S., Pachter, L., Mesirov, J. *et al.* (2000), 'Human and mouse gene structure: comparative analysis and application to exon prediction', *Genome Res.*, Vol. 10, pp. 950–958.
8. Kent, W. J. and Zahler, A. M. (2000), 'Conservation, regulation, synteny, and introns in large-scale *C. briggsae–C. elegans* genomic alignment', *Genome Res.*, Vol. 10, pp. 1115–1125.
9. Dubchak, I., Brudno, M., Loots, B. *et al.* (2000), 'Active conservation of noncoding sequences revealed by three-way species comparisons', *Genome Res.*, Vol. 10, pp. 1304–1306.
10. Jareborg, N., Birney, E. and Durbin, R. (1999), 'Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs', *Genome Res.*, Vol. 9, pp. 815–824.
11. Stojanovic, N., Florea, L., Riemer, C. *et al.* (1999), 'Comparison of five methods for finding conserved sequences in multiple alignments of gene regulatory regions', *Nucleic Acids Res.*, Vol. 27, pp. 3899–3910.
12. Gelfand, M., Koonin, E. and Mironov, A. (2000), 'Prediction of transcription regulatory sites in Archaea by a comparative genomic approach', *Nucleic Acids Res.*, Vol. 28, pp. 695–705.
13. Couronne, O., Poliakov, A., Bray, N. *et al.* (2003), 'Strategies and tools for whole-genome alignments', *Genome Res.*, Vol. 13, pp. 73–80.
14. Frazer, K., Elnitski, L., Church, D. *et al.* (2003), 'Cross-species sequence comparisons: A review of methods and available resources', *Genome Res.*, Vol. 13, pp. 1–12.
15. Tagle, D., Koop, B., Goodman, M. *et al.* (1988), 'Embryonic epsilon and gamma globin genes of a prosimian primate (*Galago crassicaudatus*). Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints', *J. Mol. Biol.*, Vol. 203, pp. 439–455.
16. Read, T.D., Salzberg, S.L., Pop, M. *et al.* (2002), 'Comparative genome sequencing for discovery of novel polymorphisms in *Bacillus anthracis*', *Science*, Vol. 296, Issue 5575, 2028–2033 (URL: http://www.sciencemag.org/cgi/content/full/296/5575/2028).
17. Volokhov, D., Rasooly, A., Chumakov, K. and Chizhikov, V. (2000), 'Identification of *Listeria* species by microarray-based assay', *J. Clin. Microbiol.*, Vol. 40, p. 4720–4728.
18. Alm, R., Ling, L. S., Moir, D. T. *et al.* (1999), 'Genomic-sequence comparison of two unrelated isolates of the human gastric pathogen *Helicobacter pylori*', *Nature*, Vol. 397, pp. 176–180.
19. Read, T., Brunham, R. C., Shen, C. *et al.* (2000), 'Genome sequences of *Chlamydia trachomatis* MoPn and *Chlamydia pneumoniae* AR39', *Nucleic Acids Res.*, Vol. 28, pp. 1397–1406.
20. Hayashi, T., Makino, K., Ohnishi, M. *et al.* (2001), 'Complete genome sequence of enterohemorrhagic *Escherichia coli* O157:H7 and genomic comparison with a laboratory strain K-12', *DNA Res.*, Vol. 8, pp. 11–22.
21. Perna, N., Plunkett, G. 3rd, Burland, V. *et al.* (2001), 'Genome sequence of enterohaemorrhagic *Escherichia coli* O157:H7', *Nature*, Vol. 409, pp. 529–533.
22. Ogata, H., Audic, S., Renesto-Audiffren, P. *et al.* (2001), 'Mechanisms of evolution in *Rickettsia conorii* and *R. prowazekii*', *Science*, Vol. 293, pp. 2093–2098.
23. Glaser, P., Frangeul, L., Buchrieser, C. *et al.* (2001), 'Comparative genomics of *Listeria* species', *Science*, Vol. 294, pp. 849–852.
24. Deng, W., Burland, V., Plunkett, G. 3rd *et al.* (2002), 'Genome sequence of *Yersinia pestis* KIM', *J. Bacteriol.*, Vol. 16, pp. 4601–4611.
25. Paulsen, I. T., Seshadri, R., Nelson, K. E. *et al.* (2002), 'The *Brucella suis* genome reveals

fundamental similarities between animal and plant pathogens and symbionts', *Proc. Natl Acad. Sci. USA*, Vol. 20, pp. 13148–13153.

26. Gardner, S., Kuczmarski, T. A., Vitalis, E. A. and Slezak, T. (2003), 'Limitations of TaqMan® PCR for detecting divergent viral pathogens illustrated by Hepatitis A, B, C and E viruses and Human Immunodeficiency Virus', *J. Clin. Microbiol.* (in press).
27. Needleman, S. and Wunsch, C. (1970), 'A general method applicable to the search for similarities in the amino acid sequence of two proteins', *J. Mol.Biol.*, Vol. 48, pp. 443–453.
28. Smith, T. and Waterman, M. (1981), 'Identification of common molecular subsequences', *J. Mol Biol.*, Vol. 147, pp. 195–197.
29. Timelogic (URL: http://www.timelogic.com).
30. Paracel (URL: http://www.paracel.com).
31. Morgenstern, B. (1999), 'DIALIGN 2: Improvement of the segment-to-segment approach to multiple sequence alignment', *Bioinformatics*, Vol. 15, pp. 211–218.
32. Vincens, P., Buffat, L., Andre, C. *et al.* (1998), 'A strategy for finding regions of similarity in complete genome sequences', *Bioinformatics*, Vol. 14, pp. 715–725.
33. Delcher, A., Kasif, S., Fleischmann, R. *et al.* (1999), 'Alignment of whole genomes', *Nucleic Acids Res.*, Vol. 27, pp. 2369–2376.
34. Schwartz, S., Zhang, Z., Frazer, K. *et al.* (2000), 'PipMaker – a web server for aligning two genomic DNA sequences', *Genome Res.*, Vol. 10, pp. 577–586.
35. Buhler, J. (2001), 'Efficient large-scale sequence comparison by locality-sensitive hashing', *Bioinformatics*, Vol. 17, pp. 419–428.
36. Kurtz, S. (2003), 'A Time and Space Efficient Algorithm for the Substring Matching Problem', Technical Report, Zentrum für Bioinformatik, Universität Hamburg.
37. Höhl, M., Kurtz, S. and Ohlebusch, E. (2002), 'Efficient multiple genome alignment', *Bioinformatics*, Vol. 18(Suppl. 1), pp. S312–S320.
38. Dumas, J. and Ninio, J. (1982), 'Efficient algorithms for folding and comparing nucleic acid sequences', *Nucleic Acids Res.*, Vol. 10, pp. 197–206.
39. Oggioni, M. and Pozzi, G. (2001), 'Comparative genomics for identification of clone-specific sequence blocks in *Streptococcus pneumoniae*', *FEMS Microbiol. Lett.*, Vol. 200, pp. 137–143.
40. Altschul, S. F., Madden T. L., Schäffer A. A. *et al.* (1997), 'Gapped BLAST and PSI-BLAST: A new generation of protein database search programs', *Nucleic Acids Res.*, Vol. 25(17), pp. 3389–3402.
41. Crossmatch (URL: http://www.phrap.org/).
42. Lee, I., Westaway, D., Smit, A. *et al.* (1998), 'Complete genomic sequence and analysis of the prion protein gene region from three mammalian species', *Genome Res.*, Vol. 8, pp. 1022–1037.
43. CGAT (URL: http://inertia.bs.jhmi.edu/CGAT/CGAT.html).
44. Schwartz, S., Miller, W., Yang, C.-M. and Hardison, R. (1991), 'Software tools for analyzing pairwise alignments of long sequences', *Nucleic Acids Res.*, Vol. 19, pp. 4663–4667.
45. Sonnhammer, E. and Durbin, R. (1995), 'A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis', *Gene*, Vol. 167, pp. GC1–10.
46. Duret, L., Gasteiger, E. and Perriere, G. (1996), 'LALNVIEW: A graphical viewer for pairwise sequence alignments', *Comput. Appl. Biosci.*, Vol. 12, pp. 507–510.
47. Galili, N., Baldwin, H., Lund, J. *et al.* (1997), 'A region of mouse chromosome 16 is syntenic to the DiGeorge, velocardiofacial syndrome minimal critical region', *Genome Res.*, Vol. 7, pp. 17–26.
48. Gottgens, B., Gilbert, J., Barton, L. *et al.* (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.*, Vol. 11, pp. 87–97.
49. Florea, L., Riemer, C., Schwartz, S. *et al.* (2000), 'Web-based visualization tools for bacterial genome alignments', *Nucleic Acids Res.*, Vol. 28, pp. 3486–3496.
50. Jareborg, N. and Durbin, R. (2000), 'Alfresco – a workbench for comparative genomic sequence analysis', *Genome Res.*, Vol. 10, pp. 1148–1157.
51. Kent, W. and Zahler, A. (2000), 'The intronerator: Exploring introns and alternative splicing in *Caenorhabditis elegans*', *Nucleic Acids Res.*, Vol. 28, pp. 91–93.
52. Mayor, C., Brudno, M., Schwartz, J. *et al.* (2000), 'VISTA: Visualizing global DNA sequence alignments of arbitrary length', *Bioinformatics*, Vol. 16, pp. 1046–1047.
53. ACT (URL: http://www.sanger.ac.uk/Software/ACT/).
54. Miller, W. (2001), 'Comparison of genomic DNA sequences: solved and unsolved problems', *Bioinformatics*, Vol. 17, pp. 391–397.
55. Morgenstern, B., Dress, A. and Werner, T.

(1996), 'Multiple DNA and protein sequence alignment based on segment-to-segment comparison', *Proc. Natl Acad. Sci. USA*, Vol. 93, pp. 12098–12103.

56. Gottgens, B., Barton, L. M., Gilbert, J. G. *et al.* (2000) 'Analysis of vertebrate SCL loci identifies conserved enhancers', *Nat. Biotechnol.*, Vol. 18, pp. 181–186.
57. Morgenstern, B., Rinner, O., Abdeddaim, S. *et al.* (2002), 'Exon discovery by genomic sequence alignment', *Bioinformatics*, Vol. 18, pp. 777–787.
58. Delcher, A. L., Phillippy, A., Carlton, J. and Salzberg, S. L. (2002), 'Fast algorithms for large-scale genome alignment and comparison', *Nucleic Acids Res.*, Vol. 30(11), pp. 2478–2483.
59. Schwartz, S., Kent, W. J., Smit, A. *et al.* (2003), 'Human–mouse alignments with BLASTZ', *Genome Res.*, Vol. 13, pp. 103–107.
60. Durbin, R., Eddy S., Krogh A. and Mitchison G. (1998), 'Biological Sequence Analysis', Cambridge University Press, Cambridge.
61. Abouelhoda, M. I., Kurtz, S. and Ohlebusch, E. (2002), 'The enhanced suffix array and its applications to genome analysis', in 'Proceedings of the Second Workshop on Algorithms in Bioinformatics', Lecture Notes in Computer Science 2452, Springer-Verlag, Berlin, pp. 449–463.
62. Zhang, Z., Schwartz, S., Wagner, L. and Miller, W. (2000), 'A greedy algorithm for aligning DNA sequences', *J. Comp. Biol.*, Vol. 7(1/2), pp. 203–214.
63. Kurtz, S., Choudhuri, J. V., Ohlebusch, E. *et al.* (2001), 'REPuter: The manifold applications of repeat analysis on a genomic scale', *Nucleic Acids Res.*, Vol. 29(22), pp. 4633–4642.
64. Zhang, Z., Raghavachari B. R., Hardison R. C. and Miller W. (1994), 'Chaining multiple alignment blocks', *J. Comp. Biol.*, Vol. 1(3), pp. 217–226.
65. Bentley, J. L. (1990), '*K-d* trees for semidynamic point sets', in 'Proceedings of the 6th Annual ACM Symposium on Computational Geometry', ACM, New York, pp. 187–197.
66. 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.*, Vol. 22, pp. 4673–4680.
67. Sanger Centre Software (URL: http://www.sanger.ac.uk/Software/).
68. Bray, N., Dubchak, I. and Pachter, L. (2003) 'AVID: A global alignment program', *Genome Res.*, Vol. 13, pp. 97–102.
69. Sonnhammer, E. and Durbin, R. (1994), 'A workbench for large scale sequence homology analysis', *Comput. Appl. Biosci.*, Vol. 10, pp. 301–307 (URL: http://www.cgr.ki.se/cgr/groups/sonnhammer/MSPcrunch.html).
70. Rutherford, K., Parkhill, J., Crook, J. *et al.* (2000), 'Artemis: Sequence visualisation and annotation', *Bioinformatics*, Vol. 16(10), pp. 944–945.
71. DisplayMUMs (URL: http://www.tigr.org/software/displaymums/).
72. MGA (URL: http://bibiserv.techfak.uni-bielefeld.de/mga/).
73. Blue Gene/L (URL: http://www.llnl.gov/llnl/06news/NewsReleases/2002/NR-02-11-08.html).

APPENDIX: AVAILABILITY OF ALGORITHMS AND VIEWERS

Alignment algorithms	
ASSIRC	Program only: ftp://ftp.biologie.ens.fr/pub/molbio/
DIALIGN	Program and server: http://bibiserv.TechFak.Uni-Bielefeld.DE/dialign/
MUMmer	Program only: http://www.tigr.org/software/mummer/
PipMaker/BlastZ	Server only: http://bio.cse.psu.edu/pipmaker/
GLASS	Program and server: http://crossspecies.lcs.mit.edu/
WABA	Program and server: http://www.soe.ucsc.edu/~kent/xenoAli/or http://www.cse.ucsc.edu/~kent/xenoAli/
LSH-ALL-PAIRS	Not available on the Internet, must contact the author at: jbuhler@cs.washington.edu
Vmatch	http://www.vmatch.de
MGA	http://bibiserv.techfak.uni-bielefeld.de/mga/
Comparative alignment viewers	
PipMaker/BlastZ	Server only: http://bio.cse.psu.edu/pipmaker/
Alfresco	Program and server: http://www.sanger.ac.uk/Software/Alfresco/
Intronerator	Server only: http://www.cse.ucsc.edu/~kent/intronerator/
VISTA	Program and server: http://www-gsd.lbl.gov/vista/
SynPlot	Program only: http://www.sanger.ac.uk/Users/jgrg/SynPlot/
ACT	Program only: http://www.sanger.ac.uk/Software/ACT/
DisplayMUMS	Program only: http://www.tigr.org/software/displaymums/