BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 3 2011, pages 334–342
doi:10.1093/bioinformatics/btq665

Sequence analysis

Advance Access publication December 9, 2010

Mugsy: fast multiple alignment of closely related whole genomes

Samuel V. Angiuoli[1,2,*] and Steven L. Salzberg[1]

[1]Center for Bioinformatics and Computational Biology, University of Maryland, College Park and [2]Institute for Genome Sciences, University of Maryland School of Medicine, Baltimore, MD, USA

Associate Editor: Dmitrij Frishman

ABSTRACT

Motivation: The relative ease and low cost of current generation sequencing technologies has led to a dramatic increase in the number of sequenced genomes for species across the tree of life. This increasing volume of data requires tools that can quickly compare multiple whole-genome sequences, millions of base pairs in length, to aid in the study of populations, pan-genomes, and genome evolution.

Results: We present a new multiple alignment tool for whole genomes named Mugsy. Mugsy is computationally efficient and can align 31 *Streptococcus pneumoniae* genomes in less than 2 hours producing alignments that compare favorably to other tools. Mugsy is also the fastest program evaluated for the multiple alignment of assembled human chromosome sequences from four individuals. Mugsy does not require a reference sequence, can align mixtures of assembled draft and completed genome data, and is robust in identifying a rich complement of genetic variation including duplications, rearrangements, and large-scale gain and loss of sequence.

Availability: Mugsy is free, open-source software available from http://mugsy.sf.net.

Contact: angiuoli@cs.umd.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on June 23, 2010; revised on November 29, 2010; accepted on November 30, 2010

1 INTRODUCTION

There are numerous sequenced genomes from organisms spanning across the tree of life. This number of genomes is expected to continue to grow dramatically in coming years due to advances in sequencing technologies and decreasing costs. For particular populations of interest, many individual genomes will be sequenced to study genetic diversity. The Cancer Genome Atlas, 1000 Genomes Project and the Personal Genome Project will generate genome sequences from at least several thousand people. For bacterial genomes, there are already over one thousand complete bacterial genomes in public databases. Often, a pan-genome concept is needed to describe a species or population (Medini *et al.*, 2005), requiring multiple sequenced genomes from the same species. There are already nine bacterial species with ten or more sequenced genomes in a recent version of RefSeq. Hundreds of individual sequenced genomes are expected for some medically relevant species and model organisms, such as *Escherichia coli*. Many of these genomes will be in the form of "draft" genomes, where the sequencing reads are assembled into numerous contigs that together represent a fraction of the actual genome, but are incomplete and contain physical sequencing gaps. In order to make use of this explosive growth in the number of sequenced genomes, the scientific community requires tools that can quickly compare large numbers of long and highly similar sequences from whole genomes.

Whole-genome alignment has become instrumental for studying genome evolution and genetic diversity (Batzoglou, 2005; Dewey and Pachter, 2006). There are a number of whole-genome alignment tools that can align multiple whole genomes (Blanchette *et al.,* 2004; Darling *et al*., 2004; Dubchak *et al.*, 2009; Hohl *et al.,* 2002; Paten *et al.*, 2008). Whole-genome alignment tools are distinguished from collinear multiple sequence alignment tools, such as tools of (Bradley *et al*., 2009; Edgar, 2004; Thompson *et al.*, 1994), in that they can align very long sequences, millions of base pairs in length, detecting the presence of rearrangements, duplications, and large-scale sequence gain and loss. The resulting alignments can be utilized to build phylogenies, determine orthology, find recently duplicated regions, and identify species-specific DNA. For divergent sequences, alignment accuracy is difficult to assess and popular methods can disagree, such was demonstrated by the relatively low level of agreement between outputs for the ENCODE regions (Chen and Tompa, 2010; Margulies *et al.*, 2007). Given the difficulties in assessing accuracy, recent development has included methods that are statistically motivated and show improved specificity (Bradley *et al*., 2009; Paten *et al*., 2008).

At shorter evolutionary distances with large fractions of identical sequences, there is less ambiguity in alignment outcomes. Yet, even within a bacterial species, aligning multiple genomes is not a trivial task, especially if the sequences contain rearrangements, duplications and exhibit sequence gain and loss. Also, despite relatively short chromosome lengths for bacteria, typically a few million base pairs, the computational complexity of multiple sequence alignment makes it a formidable challenge. Calculation of multiple alignments with a simple sum of pairs scoring scheme is known to be an NP-hard problem (Elias, 2006), which makes calculation of an exact solution infeasible for large inputs. Multiple genome alignment tools rely on heuristics to achieve reasonable run times.

There are numerous methods to compare a single pair of whole-genome sequences (Bray *et al.*, 2003; Schwartz *et al.*, 2003). The Nucmer and MUMmer package is a fast whole-genome alignment method that utilizes a suffix tree to seed an alignment with maximal unique matches (MUMs) (Kurtz *et al.*, 2004). The suffix tree implementation of MUMmer is especially efficient and can be both

*To whom correspondence should be addressed.

built and searched in time and space that is linear in proportion to the input sequence length.

Graph-based methods have been widely employed for pairwise and multiple alignment of long sequences (Raphael *et al.*, 2004; Zhang and Waterman, 2005). The segment-based progressive alignment approach implemented in SeqAn::T-Coffee (Rausch *et al.*, 2008) utilizes an alignment graph scored for consistency and a progressive alignment scheme to calculate multiple alignments. In brief, an alignment graph is composed of vertices corresponding to non-overlapping genomic regions with edges indicating matches between regions. The alignment graph can be built efficiently for multiple sequences from a set of pairwise alignments and is scored for consistency. Consistency scoring has been demonstrated to perform well in resolving problems in progressive alignment (Notredame *et al.*, 2000; Paten *et al.*, 2009). A multiple alignment can then be derived from the graph using an efficient heaviest common subsequence algorithm (Jacobson and Vo, 1992). A noteworthy property of the alignment graph is that each genomic segment that is aligned without gaps in all pairwise alignments is represented as a single vertex in the graph. This property offers an advantage for comparisons of genomes with significant sequence identity because long gap-free regions are stored as a single vertex in the alignment graph. Since the number of vertices and edges in the alignment graph is a function of the genetic diversity of the sequences and not the sequence lengths, this method allows for a compact representation and fast alignment of very long and highly similar sequences. A limitation of the SeqAn::T-Coffee tool is that it is restricted to aligning collinear sequences that are free of rearrangements.

Computational complexity is only one challenge for the comparison of numerous whole genomes. Alignment tools must handle a rich complement of genetic variation, including mutations, rearrangements, gain and loss events and duplications. For the purposes of this study, we are especially interested in tools that do not require a reference genome and can readily accept mixtures of completed and assembled draft genome data. The requirement for a single reference genome is not always practical given sampling and intra-species diversity (Deloger *et al.*, 2009). Among current tools, Enredo-Pecan (Paten *et al.*, 2008) and MLAGAN (Dubchak *et al.*, 2009) are the only ones that both report duplications and do not require a reference genome. The Threaded Blockset aligner (TBA) (Blanchette *et al.*, 2004) also does not require a reference genome for calculating the alignment, but it produces many short local alignments that require ordering against a reference genome. Progressive Mauve (Darling *et al.*, 2004, 2010) utilizes MUMs and does not require a reference; however, Mauve does not currently report duplications. M-GCAT is a whole-genome alignment tool that also utilizes MUMs and has been shown to be computationally efficient for the alignment of closely related genomes (Treangen and Messeguer, 2006) but is biased towards a reference genome.

In this article, we present a new whole-genome alignment tool, named Mugsy, which can rapidly align DNA from multiple whole genomes on a single computer. We demonstrate the performance of Mugsy on up to 57 bacterial genomes from the same species and the alignment of chromosomes from multiple human genomes. Mugsy can align draft genome sequences and does not require a reference genome for calculating the alignment or interpretation of output. Mugsy integrates the fast whole-genome pairwise aligner, Nucmer, for identifying homology, including rearrangements and duplications, with the segment-based multiple alignment method provided by the SeqAn C++ library. Mugsy also implements a novel algorithm for identifying locally collinear blocks (LCBs) from an alignment graph. The LCBs represent aligned regions from two or more genomes that are collinear and free of rearrangements but may also contain segments that lack homology and introduce gaps in the alignment. Mugsy is run as a single command line invocation that accepts a set of multi-FASTA files, one per genome and outputs a multiple alignment in MAF format. The Mugsy aligner is open source software and available for download at http://mugsy.sf.net.

2 METHODS

The Mugsy alignment tool is comprised of four primary steps (Fig. 1):

(1) an all-against-all pairwise alignment using Nucmer, refined with delta-filter (Kurtz *et al.*, 2004);

(2) construction of an alignment graph and refinement (Rausch *et al.*, 2008) using SeqAn (Doring *et al.*, 2008);

(3) identification of LCBs in the graph using code we developed; and

(4) calculation of a multiple alignment for each LCB using SeqAn::TCoffee (Rausch *et al.*, 2008).

Mugsy includes a Perl wrapper script that runs all the steps. The primary input consists of one file per genome, which may contain more than one sequence for draft genomes (i.e. a multi-FASTA file). The SeqAn library provided functions to build an alignment graph from pairwise alignments. We made three extensions to the alignment graph approach that enabled us to use it for whole-genome alignments with rearrangements and genome flux. First, we utilized the pairwise alignments from Nucmer to define the segments allowing for gaps and mismatches. Second, we modified the data structure of the alignment graph to store the orientation between matching segments so that we could detect inversions. Lastly, we implement a novel method for calculating locally collinear subgraphs from the input alignment graph. These subgraphs represent LCBs and can correspond to inversions and regions that have been gained or lost in a subset of genomes.

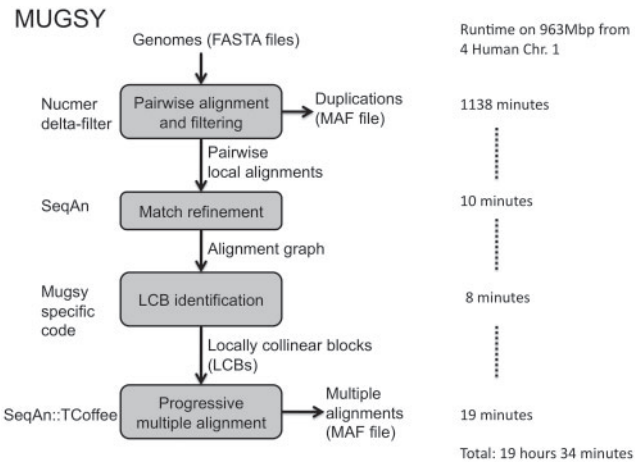

Fig. 1. The process flow and primary steps of Mugsy. The key steps are listed in boxes and data types that are input and output at each step are shown adjacent to the arrows. Software used to implement parts of each step is listed on the left. The execution time of each step from an alignment of 4 human chromosomes is provided on the right. The component timings include parsing input and writing outputs. Tests were run on a single CPU of an Intel Xeon 5570 processor with 16 GB of RAM.

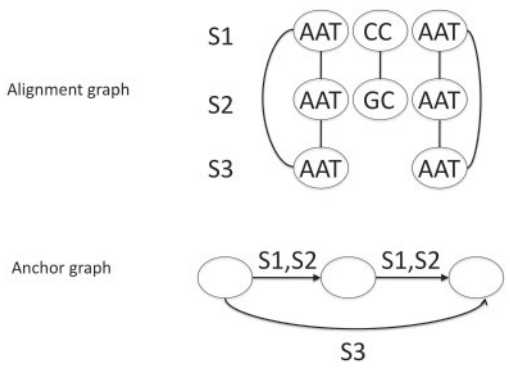

Fig. 2. Generation of multi-genome anchors from connected components in the alignment graph. Three sequences are shown (S1, S2, S3) with matching segments from the alignment graph (top). Connected components define three multi-genome anchors (bottom). Adjacent anchors along a sequence are connected by edges and labeled with the sequence identifier. To handle inconsistencies in the alignment graph, connected components are built in a greedy fashion traversing the most consistent edges first and restricting anchors to one alignment segment per genome (data not shown). Multiple segments from the same genome are allowed only if they are within a configurable distance along the sequence.

2.1 Pairwise alignment and identification of duplications

The input genomes are searched using Nucmer in an all-against-all manner using a minimum match length of 15 nucleotides and a cluster length of 60 (-l 15, -c 60). Each pairwise search is subsequently processed with the 'delta-filter' utility to identify matches likely to be orthologous. Delta-filter, a program included with Nucmer, limits pairwise matches to those contained in the highest scoring chain of matches calculated using a modified longest increasing subsequence (LIS) (Gusfield, 1997). Each match is given a score corresponding to the match length multiplied by the square of the pairwise sequence identity. Pairwise matches that are present in the LIS chain for both the reference and query sequences (delta-filter -1) are saved for use in the multiple alignment and can include inversions. This filtering is critical for excluding homology to repetitive sequences. The output of delta-filter is converted to MAF format for subsequent processing.

We modified the source code of delta-filter to report duplicated segments that are present in the LIS chain of either the reference or the query genome, but not both (delta-filter –b). The duplicated segments identified for each pairwise alignment are saved as an output file in MAF format. The chaining algorithm in delta-filter is similar to Supermap which has been used to identify orthologous segments in the presence of duplications (Dubchak *et al.*, 2009).

Following Nucmer and delta-filter, the remaining pairwise alignments are passed to the mugsyWGA program for multiple alignment. mugsyWGA first builds an alignment graph using the refinement approach described in SeqAn::T-Coffee (Rausch *et al.*, 2008), with the addition that the orientation of the alignment between segments is also saved. The alignment graph stores all the pairwise homology information calculated by Nucmer. Each vertex represents an ungapped genomic segment (Fig. 2, top). Edges represent pairwise homology statements from Nucmer that pass the orthology filtering criteria from delta-filter as described above. The refinement procedure produces a minimal subdivision of segments from all pairwise comparisons ensuring the segments are non-overlapping. We modified the alignment graph to store the relative orientation of the matches as reported by Nucmer for each edge. The alignment graph is then processed to identify LCBs.

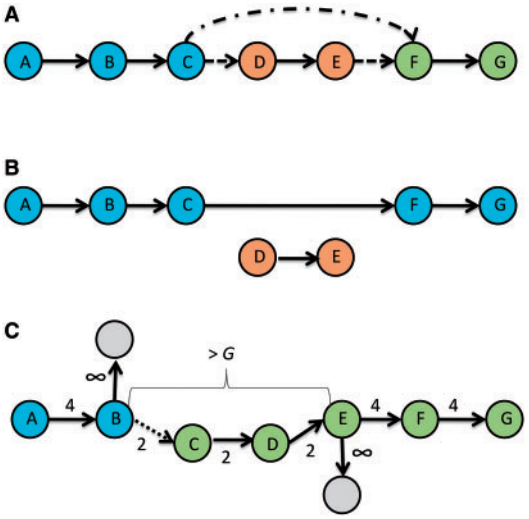

Fig. 3. Identification of LCBs in the anchor graph. A set of multi-genome anchors labeled A–G are shown. Anchors adjacent along one or more sequences are connected by an edge. (**a**) Simple paths with exactly one incoming and outgoing edge correspond to collinear regions and branches correspond to syntenic breakpoints (dotted edges) resulting in three collinear regions colored blue, orange, green. (**b**) Merging of adjacent regions. A short component (D, E) with a genomic extent less than a configurable parameter *L* is removed from the graph. The remaining anchors form a single collinear region colored blue. (**c**) Cutting of paths that violate LCBs constraints with max-flow, min-cut. Anchors B and E are adjacent but non-syntenic separated by a genomic extent greater than the configurable parameter *G* in at least one sequence. The graph forms a single connected component that is an invalid LCB. To resolve this, the anchor graph is interpreted as a flow network. Edges are labeled with an edge capacity indicating the number of sequences for which the incident anchors are collinear. Source and sink vertices (grey) are added to the graph incident to vertices that violate the distance criteria. Maximum flow, minimum cut identifies the cut (dotted edge B, C) to produce two collinear regions colored blue and green. Max-flow, min-cut ensures the graph is cut to produce collinear regions that fulfill the distance constraint *G* regardless of cycles or branches in the graph.

2.2 Determination of LCBs

A critical step in whole-genome alignment is the determination of genomic regions that are homologous, collinear, free of rearrangements and suitable for multiple alignment. Following the terminology of Mauve (Darling *et al.*, 2004), we refer to these segments as LCBs. Chaining procedures are widely utilized to define genomic intervals that are consistently ordered and oriented in multiple genomes and are often labeled as syntenic (Bourque *et al.*, 2004; Dewey, 2007; Dubchak *et al.*, 2009; Kent *et al.*, 2003; Paten *et al.*, 2008; Pevzner and Tesler, 2003). In Mugsy, we implement a new graph-based chaining procedure that looks for LCBs in the alignment graph and has similarities with previous methods for defining syntenic regions. The procedure uses heuristics to define collinear regions that are free of rearrangements and large gaps, correspond to LCBs, and are suitable for multiple alignment. The procedure first builds a graph, termed the anchor graph (Fig. 2, bottom), that enables easy identification of collinear regions by traversing simple paths comprised of anchors with exactly two incident edges (Fig. 3a).

Micro-rearrangements and repetitive elements limit the length of these regions by introducing breakpoints in the graph. Our method attempts to extend these regions by a series of merges and filtering of short LCBs (Fig. 3b). Our construction of the anchor graph joins anchors if any two genomes comprising the anchor are syntenic. This does not ensure all paths in

the graph correspond to LCBs because of genome gain, loss, duplications and rearrangements. To resolve this, a cutting procedure is used to ensure LCBs do not traverse large-scale rearrangements and indels. The cutting procedure interprets the anchor graph as a flow network and a maximum flow, minimum cut algorithm is used to trim edges from the graph to define LCBs (Fig. 3c). This procedure breaks the anchor graph at locations of reduced synteny and limits the length of an insertion or deletion described within an LCB.

The procedure takes two input parameters, a maximum genomic distance between adjacent anchors, *G*, and a minimum block length, *L*. The method will not identify rearrangements, including inversions, shorter than *L*. *G* and *L* are set in Mugsy using—distance and—minlength with defaults 1000 and 30 nucleotides, respectively. The default settings were determined empirically by varying options and comparing output to other tools on limited test data (Supplementary Figs S2–4). Increasing the value of *G* can help avoid fragmentation of LCBs in comparisons of divergent genomes but only had slight effect on datasets in this article (Supplementary Fig. S2). In alignments of 11 *Streptococcus pneumoniae* genomes, the aligned core varied by 1904 nucleotides out of ∼1.59 M core nucleotides aligned for values of *G* between 1000–10 000. In the same experiment, the total aligned nucleotides varied by 141 898 out of ∼63.3 M nucleotides. The value of *L* can have a greater impact on results with larger values excluding short regions of homology that cannot be chained into LCBs leading to reduced sensitivity.

2.3 Identification of multi-genome anchors

The first step in determining LCBs consists of producing a set of multi-genome anchors from the alignment graph. To simplify identification of synteny, we are interested in defining anchors with a single location per genomic sequence. The anchors will be subsequently chained together to define syntenic regions. The pairwise alignments used to define segments in the anchor graph have already been filtered for orthology (using delta-filter as described in Section 2.1) but inconsistencies between pairwise alignments arising from repeats and duplications can produce paths in the alignment graph with multiple segments from the same genome. As a result, connected components in the alignment graph may contain multiple segments from a single genome. Some of these copies may be close to each other on the genome while others are not. We identify duplications during pairwise alignment, and so we are interested in generating multi-genome anchors that contain only a single segment per genome.

These anchors are calculated using a greedy depth-first search of the alignment graph ordered by consistency score, traversing the highest consistency edges first (Fig. 2). In cases where there are inconsistencies in the anchor graph, we track the genomic extent of each connected component and only allow multiple segments from the same genome if they are separated by less than a configurable genomic distance, *−anchorwin*. The default value for *−anchorwin* is 100 nucleotides. Other copies explored during the search define new anchors or are excluded as singletons if no incident edges remain. By setting this parameter, we are able to reduce the size of the anchor graph for further processing. In the comparison of 31 *S.pneumoniae,* the number of multi-genome anchors was 264 133 using −anchorwin = 0 and 239 259 using −anchorwin = 100. With –anchorwin = 0, each inconsistency in the alignment graph introduces a new anchor and potential breakpoint in the anchor graph. Subsequent processing of the anchor graph attempts to merge anchors that are syntenic, including anchor fragments produced by inconsistencies in the alignment graph.

The relative orientations of segments that comprise an anchor are also saved during the greedy anchor traversal. For each LCB, the edge with the highest consistency score determines the relative match orientation for its incident genomic segments. Remaining edges are considered in descending order of consistency score, assigning a relative orientation based on the Nucmer alignment orientation. The resulting anchors consist of oriented genomic segments in two or more genomes that can contain mismatches, but no gaps, as provided by the alignment graph.

Anchors derived from this method can be very short since the refinement procedure used to build the alignment graph will produce segments as short as a single base per sequence, such as in the case of a single base indel. In the comparisons of closely related genomes, segments are often much longer and the alignment graph will often have significantly fewer vertices than the total number of base pairs in the genome. The alignment graph for ∼963 Mb from four human sequences of chromosome 1 consisted of 1 024 728 vertices with an average length of 868 bp and 1 450 084 edges. The connected components in this graph resulted in 185 537 multi-genome anchors. By comparison, the alignment graph for the 31-way comparison of *S.pneumoniae* strains, comprising 65.7 Mb in total, contained 2 717 087 vertices with an average length of 23 bp and 264 133 multi-genome anchors.

2.4 Identification of syntenic anchors

The multi-genome anchors are used to define vertices in a new directed graph, termed the anchor graph that is used to identify boundaries of LCBs. Edges in the anchor graph connect adjacent anchors along a genomic sequence. To determine edges, the vertices are first ordered along each of the member sequences. Anchors that are immediately adjacent on at least one sequence and separated by a genomic extent less than the configurable distance *G* are linked by an edge. The edges are labeled with the names of the sequences for which the anchors are adjacent. Simple paths through this graph, comprised of vertices with exactly two incident edges, represent runs of anchors that are consistently ordered and syntenic in two or more genomes. Branches in the graph produced by vertices with more than two edges represent breakpoints in synteny. The beginning and end of an assembled contig or changes in relative orientation between anchors also represent breakpoints. An initial set of LCBs is calculated by finding simple paths in the anchor graph that do not cross any breakpoints using a depth-first search (Fig. 3a). Some of these breakpoints will arise from micro-rearrangements, repetitive elements, or from our greedy construction of multi-genome anchors. The remaining steps of the algorithm attempt to extend the LCBs into longer regions that span these breakpoints by removing branches from the graph.

We merge LCBs that are connected by at least one edge in the anchor graph and do not traverse an inversion, indicated by a change in relative orientation between sequences in an anchor and do not introduce gaps greater than *G* in the projection along any member sequence (Fig. 3b). Next, anchors comprising short LCBs that span less than the minimum block length, *L*, are removed from the graph. A new set of LCBs is calculated after adding new edges between adjacent anchors separated by less than the genomic distance, *G*, on two or more genomes. This resulting graph can include branches between anchors that are adjacent on some genomes but not others due lineage specific rearrangements or indels. Repetitive elements can also give rise to branches and cycles in the graph that link anchors that are not syntenic.

An additional step is used to break edges in the anchor graph so that we ensure valid LCBs. This step models the anchor graph as a flow network and uses a maximum flow, minimum cut algorithm (Ford and Fulkerson, 1956) to find bottlenecks in the graph that are used to partition connected components that violate criteria for LCBs. Flow networks have been previously used in other areas of alignment, including the consistency problem in multiple alignment (Corel *et al.*, 2010). To build the flow network, the LCBs are ordered on each member sequence and checked for gaps greater than distance *G* or paths that join multiple contigs from the same genome. Sets of vertices that violate these criteria are deemed non-syntenic and added to opposing source and sink vertices in the flow network (Fig. 3c). We define the edge capacity of the network as the number of sequences for which any two incident anchors are adjacent and syntenic. We compute maximum flow, minimum cut using an implementation of the Ford-Fulkerson algorithm (Edmonds and Karp, 1972) to identify a minimum set of cut edges that partitions the graph ensuring the non-syntenic source and sink vertices are disconnected. This in turn ensures the LCBs consist of anchors that fulfill the maximum gap criteria and contain a single contig per genome in the case of draft genomes. The use of the max-flow, min-cut provides a valid partition even if multiple cuts are required to ensure a valid LCB due to branching in the anchor graph. This max-flow, min-cut procedure using conserved synteny

as the edge capacity has the property that it will attempt to split the LCB at bottlenecks represented by edges with reduced synteny.

The max-flow, min-cut, calculation accounted for ~12.5 min of 116 total minutes for the LCB identification in 57 *E.coli*. For these genomes, the anchor graph was composed of 675 780 multi-genome vertices and 1 258 603 edges.

Finally, the extent of the LCBs is determined from the coordinates of the minimum and maximum anchor coordinates on each member sequence. The subset of vertices in the alignment graph that overlap the extent and connected edges are passed to SeqAn::T-Coffee to align each LCB. The LCB identification procedure can produce overlapping LCB boundaries with the extent of the overlap determined by the distance parameter *G*. To place each anchor in exactly one LCB, the LCBs are sorted by length in descending order and anchors are removed from the anchor graph as they are aligned into LCBs.

The resulting multiple alignments are saved in MAF format for each LCB. The construction of the alignment graph and progressive alignment algorithm using SeqAn::TCoffee is implemented in C++ using the SeqAn library (Doring *et al.*, 2008). The LCB identification procedure is written in C++ using the Boost library (http://boost.org).

2.5 Evaluation of whole-genome alignment tools

To compare Mugsy to other multiple whole-genome alignment tools, we downloaded Mauve, TBA, FSA, MLAGAN and Pecan from their project web sites. The MLAGAN/SLAGAN and Pecan/Enredo tools do not provide scripts that automate all of the steps required to generate whole-genome alignments from a set of input FASTA files. Also, previous analyses of mammalian genomes using these tools in (Dubchak *et al.*, 2009; Paten *et al.*, 2008) utilized a compute grid to execute the pairwise alignment step. This makes generation of whole-genome alignments from a set of genomic FASTA files cumbersome. To compare these tools with Mugsy on a single computer, we limited our evaluation to only the collinear alignment components, MLAGAN and Pecan, and used Mugsy to define a common set of LCBs for evaluation. The extents of the LCBs were first calculated by Mugsy and saved as multi-FASTA files that were passed as input to MLAGAN or Pecan. MLAGAN and Pecan were run with default parameters. We did not attempt to execute the SLAGAN that defines collinear regions for MLAGAN.

Mugsy LCBs were also used to define the genomic extent of the regions passed to the multiple alignment program FSA. FSA was run using the recommended fast alignment options –fast, -noindel2, -refinement. Mugsy includes an option to invoke the FSA aligner on each LCB as a part of a post-processing step.

Mauve alignments were generated directly from the genomic FASTA files using progressiveMauve 64-bit binary version 2.3.1 with default command line options (Darling *et al.*, 2010). The Mauve output format was converted to MAF format to compare with the outputs of Mugsy.

TBA was run with default options using MAF formatted pairwise alignments from Nucmer instead of BLASTZ. The Nucmer alignments were processed with delta-filter and identical to those used as inputs for Mugsy. By using the same pairwise alignments, we were able to focus our evaluation on the multiple alignment portion of Mugsy compared to TBA. The runtime values generated are the shortest successful runtime of three tests for all tools evaluated.

For comparing outputs between Mugsy, Mauve, TBA, comparisons were restricted to completed genomes to simplify projecting pairwise alignments onto a reference coordinate system. Output files were converted to MAF format if necessary. The utility 'compare' downloaded from http://www.bx.psu.edu/miller_lab was used to calculate precision, recall, and percentage agreement between alignment outputs.

In a separate analysis, we compared the extent of LCBs calculated by Mugsy with the segmentation produced by Enredo (Paten *et al.*, 2008). Enredo reports LCBs from a set of externally generated anchors that occur in two or more input genomes. We first calculated multi-genome anchors from the alignment graph of 11 completed *S.pneumoniae* genomes as described in Section 2. The set of multi-genome anchors was used as input to both Enredo and Mugsy. Enredo was run with options –min-score = 0, –min-length = 0 and –max-gap = 3000. Additional runs were performed varying –min-length between 0 and 100 and varying –max-gap-length between 1000 and 50 000 (Supplementary Figs S5 and S6).

2.6 Data sets

The *S.pneumoniae, E.coli* and *N.meningitidis* genomes were downloaded from the NCBI Entrez website (Wheeler *et al.*, 2008). The accessions and species names are provided in Supplementary Table S1. The human genome sequences were downloaded from the individual project web sites: the NCBI reference GRCh37 available from UCSC as hg19 from http://genome.ucsc.edu, the Venter genome (JCV) from http://huref.jcvi.org (Levy *et al.*, 2007), the Kim Sungjin (SJK) genome from http://koreagenome.kobic.re.kr/en/ (Ahn *et al.*, 2009) and the YanHuang project (YH) from http://yh.genomics.org.cn (Wang *et al.*, 2008). The SJK genome utilized the NCBI reference to build consensus sequences as described in (Ahn *et al.*, 2009). The *de novo* assembly of Li *et al.* (2010) was not available as a consensus scaffold that spans chromosome 1. Instead, we utilized a consensus sequence for YH from Beijing Genomics Institute that was based on the UCSC build hg18 (NCBI v36) and is available as a single scaffold spanning chromosome 1 on the project web site (http://yh.genomics.org.cn). We choose to align these sequences to demonstrate the performance of Mugsy on the multiple alignment of very long sequences.

SNVs were obtained from the personal variant tracks of UCSC browser (Rosenbloom *et al.*, 2009) and included these sources: JCV (Levy *et al.*, 2007), YH (Wang *et al.,* 2008), SJK (Ahn *et al.*, 2009) and dbSNP 130 (Sherry *et al.*, 2001). The personal variant tracks provided the variant data in a common format with coordinates on a single version of the reference genome, hg19, which was used for multiple alignment with Mugsy. This allowed for comparison of the published variants for each individual even though some of the published studies were generated on consensus sequences prior to hg19.

3 RESULTS

3.1 Alignment of multiple bacterial genomes

We computed whole-genome alignments using Mugsy and compared runtimes to other popular whole-genome alignment tools. The input genomes consisted of a mixture of completed and draft sequences with most genomes represented in multiple contigs (Table 1). Mugsy had the second fastest runtime, requiring <2 h for the alignment of 31 *Streptococcus pneumoniae* genomes and ~19 h for the alignment of 57 *E.coli* genomes (Table 2). Nucmer+TBA had the fastest total runtime on this same dataset. Mugsy and TBA were the only two tools evaluated that completed the alignment of 57 *E.coli* in <2 days of processing on a single CPU. The step in Mugsy that identifies LCBs contributed ~15 of 56 min for the *S.pneumoniae*

Table 1. Summary of genomes compared using whole-genome alignment

Organism	Number of genomes	Number of sequences	Total bases (Mb)
N.meningitidis	5	5	10.9
S.pneumoniae	31	1906	65.7
E.coli	57	4213	299.1
Human Chr I	4	4	963.2

For genomes in draft form, the total number of assembled contigs or scaffolds is provided in column 3.

Table 2. Processing time to calculate whole-genome multiple alignments using three methods

	5 *N.meningitidis*	31 *S.pneumoniae*	57 *E.coli*	4 Human Chr 1
Pairwise search (min)	3	44	435	1138
+Mugsy (min)	3	56	720	37
+TBA (min)	<1	36	381	71
Mauve v2.3.1 (min)	5	377	DNF (1)	DNF (2)

The runtime in minutes for the pairwise search includes aligning all pairs of genome sequences with Nucmer, post-processing with delta-filter and converting output formats to MAF as described in Methods. The time provided for Mugsy and TBA is the runtime for generating the multiple alignment from the pairwise search results. The time for Mauve is the total runtime. Nucmer was run with parameters MUM length −l 10, cluster length −c 60 and all other default options. Mugsy was run with parameters −distance = 1000 and −minlength = 30. Mauve and TBA were run with default options. Tests were run on a single CPU of an Intel Xeon 5570 processor with 16 GB of RAM. DNF(1): did not finish after 2 days of processing. DNF(2): generated an allocation error.

Table 3. Summary of the whole-genome multiple alignment of 31 strains of *S.pneumoniae* using three different methods

	Number of LCBs	Length core (bp)	Core LCB N50 (bp)	Nucleotides aligned
Mugsy	2394	1 590 820	2044	63 294 709
Mauve v2.31	1366	1 568 715	2759	62 714 295
Nucmer+TBA	27 075	1 475 575	705	64 698 581

Each method reports a series of alignments that correspond to LCBs. The length of the aligned core is the total number of alignment columns that contain all input genomes and no gap characters. Half of the aligned core is contained in LCBs spanning genomic regions longer than the core LCB N50 length. The total number of aligned nucleotides is obtained by counting bases aligned to at least one other genome in the multiple alignment.

multiple alignments and ∼116 of 720 min for the *E.coli* multiple alignment.

We ran additional comparisons of runtimes with MLAGAN (Dubchak *et al.*, 2009) and Pecan (Paten *et al.*, 2008) whole-genome multiple alignment tools and the collinear alignment tool FSA (Bradley *et al.*, 2009). For this comparison, a single set of LCBs was first calculated by Mugsy to define genomic extents for multiple alignment by MLAGAN, Pecan and FSA. Of these three tools, only FSA completed the alignment of all LCBs in the 57 *E.coli* genomes in <2 days of processing on a single CPU. FSA is a fast method for aligning long sequences (Bradley *et al.*, 2009) but it is restricted to aligning collinear segments that are free of rearrangements. The runtime of FSA was slightly faster (896 min) than the combined runtime of Nucmer and Mugsy (1155 min).

The alignment positions calculated by Mugsy show agreement with those reported by Mauve and TBA. We evaluated the agreement using a projection of pairwise alignments using one of the reported outputs as a hypothetical true alignment in a comparison of 11 complete genomes in the *S.pneumoniae* dataset. Mugsy alignments scored a precision and recall of 0.99, 0.99 and 0.97, 0.99 using TBA and Mauve, respectively as truth in this comparison (Supplementary Table S2).

Mugsy aligned slightly more nucleotides than Mauve in almost double the number of LCBs for the full *S.pneumoniae* dataset (Table 3). Mugsy also identified a slightly longer core alignment. The aligned core is comprised of alignment columns that contain all input genomes and no gaps. The combination of Nucmer+TBA aligned more total nucleotides but a shorter and more fragmented core (Table 3, core N50).

The length and number of aligned regions was the primary difference in output between Mugsy, Mauve, and TBA in our evaluations. Mauve produced LCBs with the longest average length (Table 3, Supplementary Fig. S1) but did not complete the alignment of the largest data sets used in this evaluation in the allotted time. In the comparison of 11 completed *S.pneumoniae* genomes, Mugsy LCBs either shared boundaries or partially overlapped all of the Mauve LCBs (Supplementary Fig. S7).

Mugsy reported longer alignments than TBA on average (Table 3, Supplementary Fig. S1). Mugsy LCBs contained all but one of the shorter TBA blocks in a comparison of 11 completed *S.pneumoniae* genomes. 76% of all TBA blocks (2128 of 2791) were fully contained within longer Mugsy LCBs (Supplementary Fig. S7). By comparison, 25% of Mugsy LCBs (20 of 77) shared identical boundaries or were spanned by longer blocks in TBA. Slightly fewer TBA blocks were contained in Mauve than Mugsy, 2078 versus 2128.

The differences in LCB composition and boundaries are also indicated by the lengths of the contained gaps (indicating an insertion or deletion event) reported by each tool. The longest gap lengths present in a LCB for Mugsy, Mauve and TBA were 31 130 bp, 34 910 bp and 177 bp, respectively in the comparison of 31 *S.pneumoniae* genomes.

To further evaluate our method, we compared the LCB identification step in Mugsy with Enredo, another graph-based method that has been demonstrated on comparisons of mammalian genomes (Paten *et al.*, 2008). Mugsy calculated a longer aligned core and incorporated more anchors into LCBs than Enredo using a set of anchors from 11 completed *S.pneumoniae* genomes. Mugsy calculated a total of 425 LCBs comprising 22 913 396 aligned nucleotides (98.6% of input) compared to 30 710 LCBs from Enredo comprising 22 451 622 aligned nucleotides (95.4%) (Supplementary Fig. S5). The Mugsy core LCBs consisted of 1 741 704 nucleotides versus 1 229 583 nucleotides with Enredo (Supplementary Fig. S6). The Mugsy LCBs were also longer than Enredo on average, with 79% (24 401 of 30 710) of Enredo LCBs sharing identical boundaries or fully contained within longer Mugsy LCBs (Supplementary Fig. S7). By comparison, 20% (88 of 425) of Mugsy LCBs shared boundaries or were fully contained in longer LCBs reported by Enredo. Increasing the distance parameter in Enredo did not improve results (Supplementary Fig. S5). The relatively short and fragmented regions reported by Enredo may be due to the composition of the multi-genome anchors used in our comparison. As described in Section 2, the multi-genome anchors vary in length and can be subdivided during the segment refinement procedure to as short as a single base. Enredo has been previously reported to work well on longer anchors (>50 bp in (Paten *et al.*, 2008))

Mugsy includes a step for building longer syntenic regions, LCBs, from shorter multi-genome anchors. The longer aligned regions simplify some downstream analysis, such as the identification of orthologous genes and mapping of annotations, thereby minimizing the need for a reference genome. Longer alignments also aid the

inspection of genomic regions that have been gained or lost and span multiple genes without requiring a reference genome. Increasing the value of the –distance parameters in Mugsy produces longer LCBs, although with slight loss of sensitivity (Supplementary Fig. S3). Our greedy method for building multi-genome anchors can introduce branches in the anchor graph in cases where there are inconsistencies in combining pairwise alignment. Our LCB identification algorithm aims to reduce this fragmentation but remains an area that can be improved. Contig boundaries will also cause fragmentation in Mugsy LCBs. As a result, introducing draft genomes will automatically increase the number of LCBs.

3.2 Alignment of multiple human genomes

To evaluate the performance of Mugsy on larger sequences, we aligned multiple individual chromosomes from human genomes. We identified four human genomes for which consensus sequences are available for each chromosome: the NCBI reference human genome build GRCh37 (hg19 at the UCSC genome browser) (IHGSC 2001), a western European individual (JCV) (Levy *et al.*, 2007), a Korean individual (SJK) (Ahn *et al.*, 2009), and a Han Chinese individual (YH). Mugsy was able to align all four copies of chromosome 1 in <1 day using a single CPU (Table 2). Mugsy computed the multiple alignments in <1 h (37 min) after completing the pairwise searches with Nucmer. The contribution of the LCB identification step in Mugsy was ∼7 min. By comparison, TBA ran in 71 min using the same pairwise alignments as input. Realignment of all the LCBs from Mugsy with the FSA aligner ran in 358 min. Three other whole-genome alignment tools evaluated (MLAGAN, Pecan and Mauve) failed to complete an alignment of the four human chromosomes in <2 days of processing time. The length of the chromosomes (>219 Mb each) and amount of repetitive DNA in the human genome makes whole-genome alignment especially challenging. The genomes were not masked for repetitive elements.

Mugsy calculated 526 LCBs on chromosome 1 with the longest LCB spanning 5.97 Mb on all four individuals. The LCBs covered 224 975 484 of 225 280 621 (99.86%) nucleotides in the NCBI reference sequence, hg19. The alignment viewer GMAJ was used to generate pairwise percent identity plots projected from the Mugsy multiple alignment (Fig. 4). The plots show variation in JCV, SJK and YH versus the reference sequence, hg19. The sequences for YH and SJK both utilized the NCBI reference in building the consensus sequences, and therefore this comparison may underrepresent the variation in these genomes. The percent identity plots indicate this possible artifact, showing relatively low variation in the comparison of YH and SJK versus hg19.

The whole-genome multiple alignments produced by Mugsy were parsed to extract variants, including mutations, insertions and deletions. SNVs were extracted from ungapped alignment columns with more than one allele and compared to published variations in the personal variant tracks of the UCSC genome browser. Many of the SNVs calculated by Mugsy are also reported at the UCSC browser or dbSNP (Table 4). Mugsy calculates variation on assembled consensus sequence and does not consider the composition or quality of the underlying sequencing reads that contributed to the assembly. We restricted the comparison to variants annotated as homozygous for the individual using the UCSC browser. Additional variation

Table 4. Number of SNVs detected by Mugsy in the multiple alignment of human chromosome 1 from four individuals, all aligned to human reference assembly hg19

Individual genomes	Mugsy SNVs	SNVs at UCSC	Recall from UCSC	Precision from UCSC or dbSNP
JCV	216 201	108 767	104 684 (90)	194 616 (90)
SJK	135 070	113 708	112 032 (98)	128 473 (95)
YH	114 871	104 590	103 106 (98)	113 641 (99)

Mugsy alignments were performed on consensus sequences of human chromosome 1 as provided by each source. SNVs were obtained from alignment columns where the consensus nucleotide in JCV, SJK or YH differed from the nucleotide in hg19. An additional filter was applied to screen out alignment columns that contained gaps within five positions on either side. Published SNVs for JCV, SJK or YH were obtained from UCSC personal variant tracks, restricted to homozygous variants where annotated. Recall (Column 4) is the number of Mugsy variants that match UCSC divided by the total number of UCSC variants. Precision measures the number of Mugsy variants that match either UCSC or dbSNP variants divided by the total number of variants reported by Mugsy. Values within parenthesis are in percent.

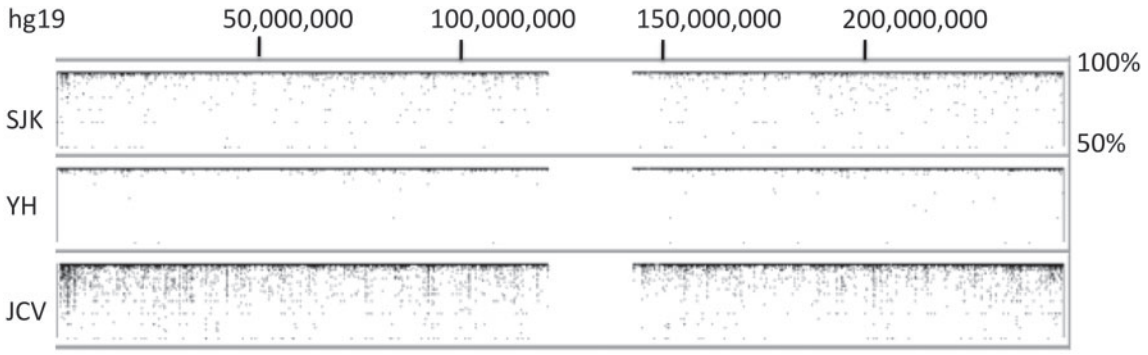

Fig. 4. Percent identity plots from the Mugsy multiple alignment of human chromosome 1 sequences of four individuals. The alignments span 99.9% of the nucleotides on chromosome 1 of the NCBI reference hg19, excluding the centromere, which is shown as a gap in the middle of the figure. The plots were obtained from the alignment viewer, GMAJ, using hg19 as reference for the display (top coordinates). A percent identity plot is displayed in subsequent rows for each of the three other genomes SJK, YH, JCV. The percent identity in each row ranges from 50 to 100 from the bottom to top of each row.

identified by Mugsy may be due to differences in detection methods or assembly artifacts in the consensus.

4 DISCUSSION

We introduce Mugsy, a new multiple whole-genome alignment tool that does not require a reference genome and can align mixtures of complete and draft genomes. Using current generation sequencing technologies, a majority of newly sequenced genomes are expected to be draft genomes represented by multiple contigs after assembly. Mugsy can identify sequence conservation and variation in any subset of these inputs.

The primary advantage of Mugsy over similar tools is speed. Mugsy is able to align 57 *E.coli* genomes (299 Mb) in <1 day on a single CPU and 31 *S.pneumoniae* genomes in <2 h. Mugsy was also the fastest tool evaluated for the alignment of four assembled human chromosomes, completing the LCB identification and multiple alignment in <1 h provided a library of pairwise alignments. Mugsy and TBA were the only tools evaluated that completed alignments of four human chromosomes and 57 *E.coli* genomes in <2 days of processing time on a single CPU. On smaller datasets of closely related genomes, we found agreement between alignments generated by Mugsy, Mauve and TBA. The primary difference was the number and boundaries of the aligned regions.

Our work relies heavily on two open source software packages, the suffix tree-based pairwise aligner Nucmer (Kurtz *et al.*, 2004) and the segment-based alignment approach of SeqAn::TCoffee (Rausch *et al.*, 2008). We utilized Nucmer to quickly build a library of pairwise homology across all input genomes. Our work extends methods in SeqAn::TCoffee to accommodate whole-genome multiple alignment with rearrangements and duplications.

Mugsy implements a new procedure that identifies LCBs. The graph utilized for the LCB identification and segment-based multiple alignment is compact for highly conserved sequences allowing for efficient computation. This makes Mugsy especially well suited to classification of species pan-genomes and other intra-species comparisons where there is a high degree of sequence conservation. Alignment of many large, highly conserved sequences, such as human chromosomes, is likely to become increasingly popular as improvements in sequencing and assembly technologies allow for de-novo assembly of human genomes, including assembly of haplotypes.

Mugsy relies on a number of parameters including minimum MUM length in Nucmer and the LCB chaining parameters. Careful choice of parameters is likely to be important for alignments at longer evolutionary distances. Automatically determining parameters or providing user guidance on parameter choice is an area that needs improvement.

Also, for more divergent genomes, the performance advantages of the segment-based alignment approach decrease as the length of the conserved segments shorten and the size of the alignment graph grows. The alignment of 57 *E.coli* strains required slightly >12 GB of RAM to build and process the alignment graph. The larger memory requirement of Mugsy on more divergent genomes is a limitation of the tool and an area that we may be able to improve at the expense of longer runtimes.

As biologists continue to explore the rich genetic diversity of the biosphere, thousands of genomes may soon be available for some species and the ability to read genetic information is outpacing the speed at which we can analyze the data for meaningful relationships. Mugsy attempts to address this performance gap but additional algorithm development is needed. The alignment of hundreds or more even relatively small bacterial genomes remains a formidable challenge and may limit the use of the growing amounts of whole-genome data by biologists. As desktop computers are now commonly available with multiple CPUs, parallel processing using multiple CPUs is a readily available technique to improve runtimes. The pairwise alignment phase of Mugsy and multiple alignment of each LCB are easily parallelized and well suited to run on multi-CPU cores or compute clusters. We plan to extend Mugsy to support parallel processing to enable alignment of even larger datasets.

Whole-genome alignment incorporates sequence identity and synteny and is well suited to aiding annotation of orthologous genes and pan-genomes. We also plan to add support for deriving these features from the output of whole-genome alignments as future work.

ACKNOWLEDGEMENTS

We thank Herve Tettelin, Jason Stahl, Dave Rasko, Florian Fricke, David Riley and the anonymous reviewers for thoughtful feedback for suggestions.

Funding: National Institutes of Health (R01-LM006845 and R01-GM083873 to SLS) in part.

Conflict of Interest: none declared.

REFERENCES

Ahn,S.M. *et al*. (2009) The first Korean genome sequence and analysis: full genome sequencing for a socio-ethnic group. *Genome Res*., **19**, 1622–1629.

Batzoglou,S. (2005) The many faces of sequence alignment. *Brief Bioinform.,* **6**, 6–22.

Blanchette,M. *et al*. (2004) Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, **14**, 708–715.

Bourque,G. *et al*. (2004) Reconstructing the genomic architecture of ancestral mammals: lessons from human, mouse, and rat genomes. *Genome Res*., **14**, 507–516.

Bradley,R.K. *et al*. (2009) Fast statistical alignment. *PLoS Comput Biol*., **5**, e1000392.

Bray,N. *et al*. (2003) AVID: A global alignment program. *Genome Res*., **13**, 97–102.

Chen,X. and Tompa,M. (2010) Comparative assessment of methods for aligning multiple genome sequences. *Nat. Biotechnol.*, **28**, 567–572.

Corel,E. *et al*. (2010) A min-cut algorithm for the consistency problem in multiple sequence alignment. *Bioinformatics*, **26**, 1015–1021.

Darling,A. *et al*. (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res*., **14**, 1394–1403.

Darling,A. *et al*. (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. *PLoS ONE*, **5**, e11147.

Deloger,M. *et al*. (2009) A genomic distance based on MUM indicates discontinuity between most bacterial species and genera. *J. Bacteriol*., **191**, 91–99.

Dewey,C.N. (2007) Aligning multiple whole genomes with Mercator and MAVID. *Methods Mol. Biol*., **395**, 221–236.

Dewey,C.N. and Pachter,L. (2006) Evolution at the nucleotide level: the problem of multiple whole-genome alignment. *Hum. Mol. Genet*., **15 (Spec No. 1)**, R51–R56.

Doring,A. *et al.* (2008) SeqAn an efficient, generic C++ library for sequence analysis. *BMC Bioinformatics,* **9**, 11.

Dubchak,I. *et al.* (2009) Multiple whole-genome alignments without a reference organism. *Genome Res*., **19**, 682–689.

Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.,* **32**, 1792–1797.

Edmonds,J. and Karp,R.M. (1972) Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, **19**, 248–264.

Elias,I. (2006) Settling the intractability of multiple alignment. *J. Comput. Biol.,* **13**, 1323–1339.

Ford,F.R. and Fulkerson,D.R. (1956) Maximal flow through a network. *Can. J. Math*., **8**, 399–404.

Gusfield,D. (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, New York.

Hohl,M. *et al.* (2002) Efficient multiple genome alignment. *Bioinformatics*, **18**, (Suppl. 1) S312–S320.

IHGSC (2001) Initial sequencing and analysis of the human genome. *Nature,* **409**, 860–921.

Jacobson,G. and Vo,K.-P. (1992) Heaviest increasing/common subsequence problems. *Proceedings of the Third Annual Symposium on Combinatorial Pattern Matching*, Springer, pp. 52–66.

Kent,W.J. *et al.* (2003) Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proc. Natl Acad. Sci. USA*, **100**, 11484–11489.

Kurtz,S. *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol*., **5**, R12.

Levy,S. *et al.* (2007) The diploid genome sequence of an individual human. *PLoS Biol*., **5**, e254.

Li,R. *et al.* (2010) De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res*., **20**, 265–272.

Margulies,E.H. *et al.* (2007) Analyses of deep mammalian sequence alignments and constraint predictions for 1% of the human genome. *Genome Res*., **17**, 760–774.

Medini,D. *et al.* (2005) The microbial pan-genome. *Curr. Opin. Genet. Dev*., **15**, 589–594.

Notredame,C. *et al.* (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol*., **302**, 205–217.

Paten,B. *et al.* (2009) Sequence progressive alignment, a framework for practical large-scale probabilistic consistency alignment. *Bioinformatics*, **25**, 295–301.

Paten,B. *et al.* (2008) Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. *Genome Res*., **18**, 1814–1828.

Pevzner,P. and Tesler,G. (2003) Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res*., **13**, 37–45.

Raphael,B. *et al.* (2004) A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res*., **14**, 2336–2346.

Rausch,T. *et al.* (2008) Segment-based multiple sequence alignment. *Bioinformatics*, **24**, i187–i192.

Rosenbloom,K.R. *et al.* (2009) ENCODE whole-genome data in the UCSC Genome Browser. *Nucleic Acids Res*., **38**(Database issue), D620–D625.

Schwartz,S. *et al.* (2003) Human-mouse alignments with BLASTZ. *Genome Res*., **13**, 103–107.

Sherry,S.T. *et al.* (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res*., **29**, 308–311.

Thompson,J.D. *et al.* (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.

Treangen,T.J. and Messeguer,X. (2006) M-GCAT: interactively and efficiently constructing large-scale multiple genome comparison frameworks in closely related species. *BMC Bioinformatics*, **7**, 433.

Wang,J. *et al.* (2008) The diploid genome sequence of an Asian individual. *Nature*, **456**, 60–65.

Wheeler,D.L. *et al*. (2008) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.,* **36**(Database issue), D13–D21.

Zhang,Y. and Waterman,M.S. (2005) An Eulerian path approach to local multiple alignment for DNA sequences. *Proc. Natl Acad. Sci. USA*, **102**, 1285–1290.