CHAINER: Software for Comparing Genomes

Mohamed Ibrahim Abouelhoda[1] *and Enno Ohlebusch*[1]

[1]*Faculty of Computer Science, University of Ulm, 89069 Ulm, Germany*

ABSTRACT

Summary: Recently, software-tools for pairwise or multiple comparison of genomic sequences have gained an enormous importance in comparative genomics. Our novel software CHAINER can be used for several comparative tasks: (1) finding regions of high similarity (candidate regions of conserved synteny), (2) multiple global alignment of whole genomes, (3) comparison of multiple draft (or finished) genomes among themselves, and (4) cDNA/EST mapping. The software is available upon request.

Contact: {mibrahim,eo}@informatik.uni-ulm.de

CHAINER performs comparison tasks by chaining fragments, i.e., segments in the genomic sequences that are similar. The fragments can be exact matches (maximal unique matches as in MUMmer (Delcher *et al.*, 2002), maximal exact matches as in MGA (Höhl *et al.*, 2002) and AVID (Bray *et al.*, 2003), or exact k-mers as in GLASS (Batzoglou *et al.*, 2000)). But, one can also allow substitutions (yielding fragments as in DIALIGN (Morgenstern *et al.*, 1998) and LAGAN (Brudno *et al.*, 2003)) or even insertions and deletions (as the BLASTZ-hits (Schwartz *et al.*, 2003) that are used in PipMaker (Schwartz *et al.*, 2000)). Figure 1 gives an overview of the software-tool. Each of the fragments has a positive score that can, for example, be the length of the fragment (in case of exact fragments) or its similarity score.

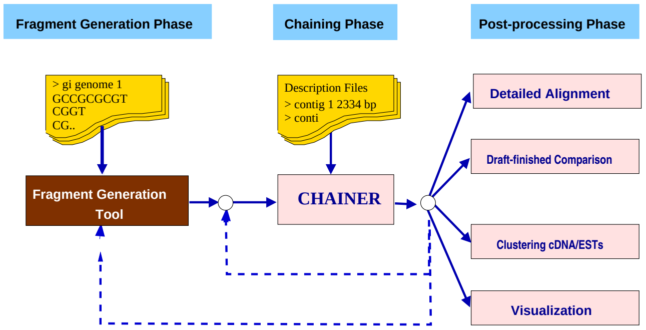

Fig. 1. The dark box is the external software for the generation of fragments. The input to the fragment generation tool are the files containing the genomes. CHAINER includes extra information in the description files such as the size of contigs and cDNA/ESTs. The feedback arrows symbolize recursive calls: It is possible to generate shorter fragments and run CHAINER again or to further chain the output chains.

FINDING REGIONS OF HIGH SIMILARITY

CHAINER delivers regions of high similarity of multiple genomes in terms of highest-scoring (*local*) chains of non-overlapping colinear fragments. A chain is just a sequence of fragments and its score is the sum of the scores of the fragments, where gaps between them are penalized. Two fragments in a chain are *colinear* if the order of their respective segments is the same in all genomes. In the pictorial representation of Figure 2(a), two fragments are colinear if the lines connecting their segments are non-crossing (e.g., the fragments 1 and 4 are colinear, while 1 and 2 are not). Two fragments *overlap* if their segments overlap in one of the genomes (e.g., the fragments 2 and 4 are overlapping, while 1 and 4 are non-overlapping). The gap cost is computed as follows. Given k genomes, let $beg(f).x_i$ and $end(f).x_i$ denote the start and end positions of a fragment f in genome $1 \leq i \leq k$. The gap cost $g(f, f')$ between two colinear fragments f and f'(i.e., $end(f).x_i < beg(f').x_i$ for all $1 \leq i \leq k$) is computed by the formula:

$$g(f, f') = \sum_{i=1}^{k} |beg(f').x_i - end(f).x_i|$$

In (Abouelhoda and Ohlebusch, 2003a), it is shown that the highest scoring chains have positive scores. Moreover, if the user sets the gap cost function to zero, then the highest scoring chain is an optimal global chain of the fragments; see the next section.

Inversions can be taken into account (an inversion is a genome rearrangement, where a section of the genome is excised, reversed in orientation, and re-inserted) by chaining fragments computed between the reverse complement of some genomes and the other genomes.

Our chaining algorithm is based on the line-sweep paradigm and uses range maximum queries (`RMQ`) with activation. During the line sweep procedure, the fragments are scanned w.r.t. their order in one of the genomes. If an end point of a fragment is scanned, then it is activated. If a start point of a fragment is scanned, then we connect it to an activated fragment of highest score occurring in the rectangular region bounded by the start point of the fragment and the origin. This highest-scoring fragment is found by an `RMQ`; see Figure 2(b). Furthermore, the user can restrict the region of the `RMQ`, which constrains

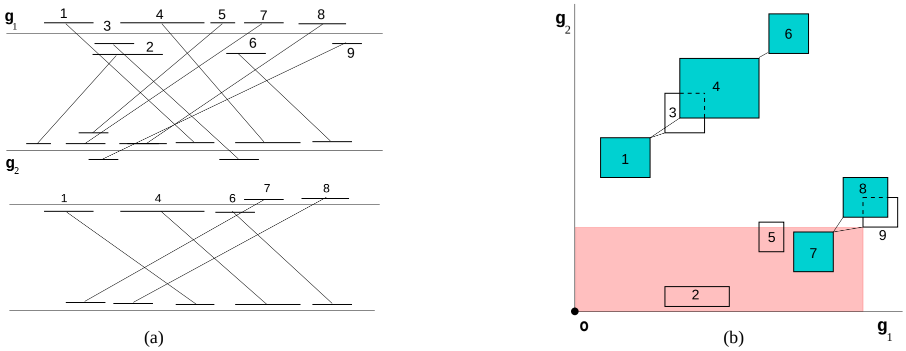

Fig. 2. Computation of highest-scoring chains of colinear non-overlapping fragments. (b) depicts the fragments as rectangles in the plane. The range of the `RMQ` at fragment 9 is bounded by the colored rectangle. The chains 1,4,6 and 7,8 are highest-scoring.

the gap length allowed between fragments. This option prevents unrelated fragments from extending the chain. Details of this algorithm, which is subquadratic in time, and experimental results can be found in (Abouelhoda and Ohlebusch, 2003a).

MULTIPLE GLOBAL ALIGNMENT

Most of the above-mentioned software-tools compute alignments of genomic sequences. To cope with the shear volume of data, they use anchor-based methods that are composed of three phases: (1) computation of fragments, (2) computation of a highest-scoring *global* chain of colinear non-overlapping fragments, and (3) alignment of the regions between the fragments of the global chain (*the anchors*) by applying the same method recursively with less stringent parameters or by using standard dynamic programming. The global chaining algorithm of CHAINER, which runs in subquadratic time, is more efficient than previous chaining algorithms; see (Abouelhoda and Ohlebusch, 2003). Moreover, it is superior to those algorithms because of the ability to incorporate gap costs. This algorithm is currently used in the multiple global alignment tool MGA (Höhl *et al.*, 2002).

COMPARING DRAFT GENOMES

Draft to draft comparison

In contrast to a finished genome, a draft genome consists of contigs of unknown order and orientation (a contig is a contiguous tract of known subsequence of the genome). Because the cost of finishing a genome can drastically exceed the cost of the draft, we anticipate massive amounts of draft microbial genomes in the near future. To cope with this data, CHAINER allows to compare a set of draft (or finished) genomes among themselves (a finished genome is considered as a draft genome consisting of a single contig). In the *direct* mode, CHAINER applies an all-against-all comparison strategy: It searches for local chains (regions of high similarity) between each contig (and its reverse complement, respectively) of a draft genome and all contigs of the remaining draft genomes. In the *high-throughput* mode, however, all draft genomes are compared to each other in a single run of the chaining program. To this end, all contigs (and their reverse complements) of a draft genome are concatenated, but a unique separator symbol is inserted between consecutive contigs to represent their border. The fragments are then generated w.r.t. the concatenated sequences and sorted according to their order in one of the concatenated sequences. This enables us to use the same line-sweep algorithm as in Section 1, but we have to make sure that the chains “do not cross the borders between contigs.” This can be done by deactivating fragments that lie in different contigs, which in turn restricts the range of a range maximum query; see Figure 3.

Draft to finished comparison

CHAINER also provides functionality for post-processing the local chains when a single draft genome is compared to a finished genome. If it is known—or strongly believed—that the finished genome is very similar to the draft (so that there are no or few rearrangements), then it is possible to determine the order and orientation of the contigs with the help of the finished genome. CHAINER has a subroutine that achieves this by first running the previous algorithm and then selecting every contig that has a local chain covering a high percentage (user-defined parameter) of the contig length. The order and orientation of these contigs in the draft genome is thus determined by the order of their occurrences in the finished genome. For the case that the finished genome is not very similar to the draft genome, CHAINER has a subroutine that identifies common regions as well as regions that are unique to either genome. This is done by finding a set of non-overlapping local chains such that the total coverage w.r.t.

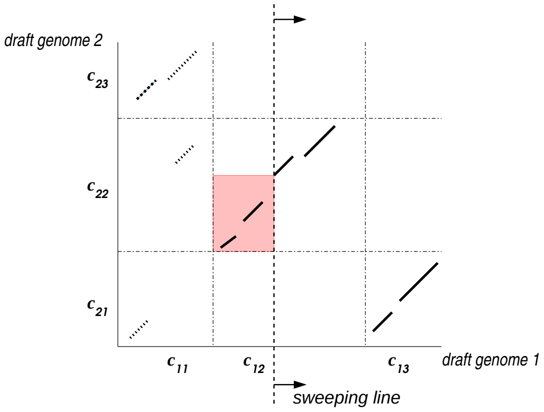

Fig. 3. The contigs c_{11}, c_{12} and c_{13} of the first draft genome are compared to the contigs c_{21}, c_{22} and c_{23} of the second draft genome. The dash-dotted lines represent the borders between the contigs. Because the fragments of contig c_{11} are deactivated, the range maximum query is restricted to the colored region.

the finished genome is maximized. The regions covered by these chains are the common ones and the remaining regions are unique to either genome.

cDNA MAPPING

An important step in annotating a newly sequenced genome, and also in gene recognition, is the mapping of cDNA/ESTs to the genome. Complementary DNA (cDNA) is obtained from mRNA through reverse transcription. When the mRNA stems from eukaryotes, the cDNA consists only of the exons of the expressed gene because the introns have been spliced out. While the exons are short segments ranging from tens to hundreds of base pairs, the introns span segments in the order of kilo-bases. Expressed sequence tags (ESTs) are segments of the cDNA usually obtained by sequencing their 3’ and 5’ ends. The problem of cDNA mapping is to find the gene and its exon/intron structure on the genome from which the cDNA originated; see Figure 4.

As in the comparison of draft genomes, CHAINER has two modes for cDNA mapping. The first is the *direct* mode which maps a single cDNA sequence to one or more genomic sequences. This option generates a global chain of fragments by the same method as in Section 2, but without penalizing the gaps between the fragments (because the gaps correspond to introns). The second mode is the *high-throughput* mode, which allows one to compare one or several genomes (or parts of them) to a complete cDNA or EST database. The implementation of this mode is similar to that of Section 3. For both modes, it is recommended to use fragments of non-exact matches so that sequencing errors are accounted for.

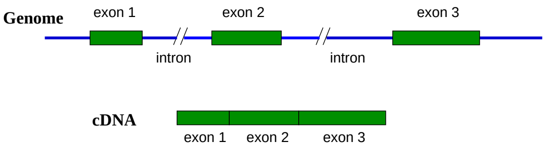

Fig. 4. An example of cDNA mapped to a genomic sequence.

CHAINER also provides functionality for post-processing the mapped cDNA sequences (or ESTs). One subroutine clusters and reports the mapped cDNA sequences whose positions are overlapping in the genome. This feature helps in detecting alternatively spliced genes. Still, this mapping is just a first step towards gene finding, and more complicated intron models, such as those used in (Florea *et al.*, 1998; Mott, 1997), should be used for further analysis.

REFERENCES

Abouelhoda,M.I. and Ohlebusch,E. (2003a) A local chaining algorithm and its applications in comparative genomics. *Proc. 3rd Workshop on Algorithms in Bioinformatics, LNBI*, **2812**, 1–16.

Abouelhoda,M.I. and Ohlebusch,E. (2003) Multiple genome alignment: Chaining algorithms revisited. *Proc. 14th Annual Symposium on Combinatorial Pattern Matching, LNCS*, **2676**, 1–16.

Batzoglou,S., Pachter,L., Mesirov,J.P., Berger,B. and Lander,E.S. (2000) Human and mouse gene structure: Comparative analysis and application to exon prediction. *Genome Research*, **10(7)**, 950–958.

Bray,N., Dubchak,I. and Pachter,L. (2003) AVID: A global alignment program. *Genome Research*, **13**, 97–102.

Brudno,M., Do,C.B., Cooper,G.M., Kim,M.F., Davydov,E.D., NISC Comparative Sequencing Program, Green,E.D., Sidow,A. and Batzoglou, S. (2003) LAGAN and Multi-LAGAN: Efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research*, **13(4)**, 721–731.

Delcher,A.L., Phillippy,A., Carlton,J. and Salzberg,S.L. (2002) Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Research*, **30(11)**, 2478–2483.

Florea,L., Hartzell,G., Zhang,Z., Rubin,G.M. and Miller,W. (1998) A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research*, **8**, 967–974.

H¨ohl,M., Kurtz,S. and Ohlebusch,E. (2002) Efficient multiple genome alignment. *Bioinformatics*, **18**, S312–S320.

Morgenstern,B., Frech,K., Dress,A. and Werner,T. (1998) DIALIGN: Finding local similarities by multiple sequence alignment. *Bioinformatics*, **14**, 290–294.

Mott,R. (1997) EST_GENOME: A program to align spliced DNA sequences to unspliced genomic DNA. *Computer Applications in the Biosciences (CABIOS)*, **13(4)**, 477–478.

Schwartz,S., Kent,W.J., Zhang,Z., Baertsch,R., Hardison,R.C., Haussler,D. and Miller,W. (2003) Human-mouse alignments with BLASTZ. *Genome Research*, **13**, 103–107.

Schwartz,S., Zhang,Z., Frazer,K.A., Smit,A., Riemer,C., Bouck,J., Gibbs,R., Hardison,R.C. and Miller,W. (2000) PipMaker–a web server for aligning two genomic DNA sequences. *Genome Research*, **10 (4)**, 577–586.