Whole Genome Alignments and Synteny Maps

IINTRODUCTION

It was not until closely related organism genomes have been sequenced that people start to think about aligning genomes and chromosomes instead of short DNA sequences. There are a large number of algorithms and methods developed for sequence alignment, in order to find conserved sequences across species. Smith & Waterman and Needleman & Wunsch algorithms are among the most popular algorithms, which uses dynamic programming to calculate scores for pair of aligned sequences and evaluate the significance. However, this technique cannot be implemented directly into aligning whole genome DNA sequences. The problem mainly lies in the difficulty in handling the large size of sequences: genome sequences may be thousands as large as single gene sequence. The traditional algorithms could require an extremely large memory to alignment two genome sequences in a reasonable period of time, which is hard to achieve. And, more importantly, these algorithms cannot distinguish paralogs in the genome and they mainly focused on sensitivity, which could generate large amount of false positives. New algorithms specifically designed for genome alignments have been developed since the late 1990s, and this paper looks into these methods, gives some opinions on synteny alignment of genome sequences.

GOALS AND CHALLENGES

Needleman & Wunsch algorithm uses classic dynamic programming method to alignment two sequences. It has $O(n^2)$ complexity, and quickly becomes unfeasible when

aligned sequences are really long like the whole genome sequence or human chromosome 1, as long as 200 Mb. Smith & Waterman algorithm uses hashing techniques in that it first finds k-mer in the sequences and does the extension by dynamic programming. Gapped BLAST which uses this technique is much faster because it only picks most outstanding matches to extend for HSPs. However, during the extension step, it also has $O(n^2)$ complexity.

These algorithms are local alignment algorithms instead of global alignments. For synteny map alignment, what we want is to find out the locations of interested orthologs in sequence pairs, and how those orthologs queued in the genome. Local alignment could give some information of ortholog positions, but extra work should be done to generate a complete genome synteny map considering tandem repeats and transposition.

New algorithms for whole genome alignment should meet the requirements listed below:

1. handle the problem with reasonable memory space and time
2. generate good synteny map alignment, well handle tandem repeats and transpositions
3. generate good local alignment in terms of specific regions in the genome

METHODS AND PERFORMANCE

Basically, variants in whole genome alignment algorithms are using strategies of either i) pairwise sequence alignment or ii) anchoring alignment. Pairwise sequence alignment is a natural extension from classic local sequence alignment using dynamic programming; however, it manages to solve the problem of memory space and time. Anchoring

alignment has a new algorithm strategy designed for long sequence alignments, free of dynamic programming. It uses a similar and generalized idea of hashing techniques used in Gapped BLAST. Basically, it finds long identical fragments in both sequences and fills the gaps between those fragments with local alignment algorithms. The methods discussed here are ordered by their publishing year.

ATGC

ATGC (Another Tool for Genome Comparison) is a multithreaded parallel implementation of dynamic programming method in pair wise sequence alignment. The basic idea is using multiple processors and threads running parallel to meet the needs of intensive computation of using dynamic programming to align genome sequences. It divides the scoring matrix into smaller rectangular blocks, assigns each of them to one thread, having two independent threads per processor. It further divides each thread into two fibers which repeatedly instantiated to make computation more efficiently. This method only report high scored alignments, because possible alignment numbers are growing exponentially. High score alignments are selected as the similarity matrix is calculated, thus the whole matrix need not to be stored. It takes 1.3 hours on a 128 processors, 128MB memory clustering machine to align *M. pneumoniae* (816394 bp) and *M. genitalium* (580074 bp).

BlastZ

BlastZ is an experimental variant of Gapped Blast. It is an implementation designed for aligning two very long sequences. And Gapped Blast algorithm is improved in such a

way that computer memory should not be a problem in implementing dynamic programming algorithm. BlastZ could only be executed on the web server called PipMaker, and unfortunately, the detailed algorithm improvements and implementation methods are not provided in the original paper.

MUMmer

MUMmer is the first anchoring alignment tool developed for alignment genome sequences. First, it computes all the MUMs (Maximal Unique Match) from two sequences, which are sequences that occurs only once in both genomes and is not contained in any longer MUM sequences. Suffix tree data structure is used to generate this decomposition. The MUM sequence is similar to the k-mer exact matching idea used in hashing technique, although here, k-mer is not restricted and could varied in length. Second, MUMs are selected and sorted, so that the longest possible set of MUMs that occur in the same order in both genomes is generated. Third, using this set as anchors and the gaps between those anchors are closed by local sequence alignments. The basic assumption behind this method is that the two genomes are closely related, so that exact matched MUMs are abundant and long enough for anchoring at genome scale. Thus, one disadvantage of MUMmer is that when not closely related genomes are aligned, it may take a lot longer to close gaps. Other disadvantages include: exact matches occurring more than once could also be meaningful; overlapping MUMs should be consistently handled; gaps larger than a certain limit remain unaligned; large exact match transposition sequences could only be aligned with dynamic programming, which is less efficient.

However, MUMmer uses the fact that closely related genomes have large amount of identical sequences to simplify alignment to exact matching, which circumvents the expense of long sequence dynamic programming alignment. And it is capable in finding SNPs, insertions or deletions, polymorphic regions in some non-coding sequences and tandem repeats. Finding these genome scale elements are important in genome alignment, and MUMmer is surely a successful method in this case.

GLASS

GLASS (Global Alignment System) is another anchoring alignment tool, although it contains dynamic programming algorithm in itself. First, it searches all pairs of exacting matching k-mers in both genomes, and the initial k is 20. Second, calculate a score for the pair in the following fashion: using dynamic programming to calculate the score of 12 nucleotides to the left and to the right, and the final score is the sum of two. Third, it removes matches with scores below a certain threshold, as well as those overlapping matches. Forth, high score matches occurring in the same order in both genomes is generated as a set, which further serves as anchors for next round of computing. Fifth, steps 1-4 is recursively applied to the gaps between the anchors with decreasing k value. Finally, close all gaps remained by standard dynamic programming.

The assumption behind the method is that strong local similarities calculated by fast local alignments are highly expected to be part of the global optimal alignment. Instead of

finding long exact matching and closing gaps once, this method actually closes gaps with shorter k-mers, recursively.

CHAOS

CHAOS (Chains Of Scores) is another method designed to find the chain of anchors. An anchor is defined as a chain of seeds combined with certain criteria. And a seed is a pair of sequences of length 7 with at least n identical base pairs. Seeds are chained together if within 20 bp distance, any other seed located within 5bp gap distance exists. This results in 2.1 anchor points per kb on average, and then more sensitive dynamic programming alignment are performed to close the gaps in between.

This method, followed by another sensitive alignment method DIALIGN to close the gaps, uses only 5% time of using DIALIGN alone. This indicates a significant improvement of using anchors to reduce calculation time.

BLAT

BLAT is not designed for cross-species DNA alignment, and it could be used directed into genome alignment. However, Couronne et al. used BLAT to find anchors in aligning two genome sequences, and they found it performed very well. First, BLAT matches are sorted by score, and those within a certain threshold is grouped together. Second, extend the groups region out in a certain fashion, and the extension should be less than 50kb. Third, these group regions pairs are score by BLAT, and those with a score less than 30% of the score from the best group are removed. Finally, the remaining groups regions serve

as anchors for global alignment. It took 17 hours to align whole human and mouse genomes on a 16 2.2GHz cluster machine.

UniMarker

UniMarker is another anchoring method generating synteny map for very large genomes as the whole human genome. First, UMs are identified by a 16-mer sliding window down the human genome and mouse genome, which occurs only once in the genome. Thus, for each genome, a library of UMs is identified. Second, in order to align genome A and B, genome B is divided into a set of overlapping fragments, each containing an equal number of UM pairs. For the human genome, a fragment has the length similar to the Y chromosome. Third, using a sliding window of 50kb and moving step of 10 kb in genome A, a scoring matrix is calculated against the fragment in B: value is the ratio of the NO of common UMs over NO of UMs in the window. Forth, if at least four consecutive windows have a score in the top 1.5% of all scores, it is identified as an anchoring island. Fifth, the opposite is performed in that genome A is fragmented and a sliding window is in genome B. This reduces the likelihood of false positives. Finally, the anchoring islands are merged into conserved regions, thus generates a synteny map.

Synteny mapping is interested in finding the conserved regions between two genomes, thus it does not require the high resolution local alignment. UniMarker uses 16-mer UMs as markers in the genome, and thus make it easier to find corresponding UMs in the other genome, which further turns to a synteny map.

DISCUSSION AND FUTURE DIRECTIONS

Aligning two very long sequences like genomes or chromosomes has major problem of memory and time if dynamic programming technique is used directly. Two strategies are developed to overcome this problem. One is straightforward, using multiple processors and high-performance servers to calculate large similarity matrix in a reasonable time. BlastZ and ATGC are in this category. The advantage of this strategy is that alignment has high resolution and high sensitivity. Thus, it does not require the aligned genomes to be very close in evolution. A proper scoring matrix could handle the evolution distance difference and the whole statistical background in local alignment could be transferred to global alignment here. However, it could still be a big problem if whole human and mouse genomes are aligned.

The other strategy is using the anchoring alignment method, which circumvents intensive computation of dynamic programming alignment for two long sequences. Anchors are exact match subsequences in the genome, thus they are most distinguishable hits in the global alignment. Different algorithms are developed to find those anchors, although an anchor defined in one method could be different from another. MUMmer uses variant lengths maximum match subsequences as anchors; GLASS uses a 20-mer match as an anchor, and has smaller-mer matches for detailed alignment; CHAOS uses chained seeds as anchors, which are not totally exact matches, but are actually highly similar sequences; BLAT anchors are not exact match sequences, either, which are mainly formed by high-scoring BLAT hits, and extended in some extent; UniMarker uses 100kb or so subsequences as anchors, which are formed by high UM density regions. With the

anchors, more sensitive and computational intensive local alignment is performed to close the gaps in between to generate whole genome alignment maps. The advantage of this strategy is that it is very space and time efficient and also meets the needs in finding conserved regions in those more than 90% similarity genomes. Later developed anchor finding algorithms could tolerate minor mismatches in anchors, while the anchor as a whole is highly conserved. This partially solves the problem that only close related genomes could be aligned with early anchoring method.

There are no "golden rules" in properly defining anchors for aligning, however, certain requirements should be considered seriously. (1) It is better to tolerate minor mismatches below a certain threshold in order to align not closely related genomes. Thus, exact match subsequences should act as cores and anchors should be a larger region that is dense with those cores. (2) Anchor regions should be in a proper size to optimize time efficiency and alignment quality. Larger anchor regions could reduce alignment time; however, unfavorable non-conserved regions could be included as false positives. On the contrary, smaller anchors could largely increase time but could result in better quality. Thus, for a specific method, different thresholds should be tested for different anchor sizes, in order to get optimized sensitivity and specificity rates.

The main goal of aligning genome sequences is to find corresponding positions of conserved sequences in two genomes. Thus UniMarker method seems to be a good choice if no high resolution alignment is needed. It is very fast, and could handle even human against mouse whole genome comparison. Synteny map is generated in order to

give information on transposition, transversion and insertion of large DNA pieces at genome scale. However, if smaller genomes are aligned to have a higher resolution, other actual alignment methods could be better.

Some of the methods, e.g. MUMmer and BLAT, have been further developed into multiple whole genome alignment algorithms in recent years. The revised idea is finding common anchors in multiple genomes, and using multiple local alignment tools like ClustalW to close the gaps.

Following are the major fields I found that could serve as future directions of whole genome alignment. (1) Establish evaluation system. It is now difficult to evaluate the performance of a whole genome alignment method. Most of the methods actually compare the alignment results to BlastZ results to get a general idea of its performance. However, the actual conserved region set itself remains an open question for human or mouse genome. It is possible to set up several testing region pairs from different organisms, such as human chromosome 20 and mouse chromosome 2, or *M. pneumoniae* and *M. genitalium* genomes. Results from different methods should be compared and evaluated in terms of sensitivity and specificity. (2) Develop visualized interface for alignment results. This requires a alignment browser to allow large-scale genome comparison view, as well as zoom in to look into detailed local alignment. UCSC genome browser gives a good example, because it listed other vertebrates' conservation situation when looking at human genes. However, a better browser for genome alignment should have "blocks" for visualize conserved regions, as well as sequences for detailed matching

information. (3) Combining existing algorithms and methods to form more efficient and competent methods. It could be interesting if MUMmer and GLASS are integrated to find anchors. We could first find MUMs, and switch to GLASS to look for 16mer exact matches, followed by 10mer, 5mer matches to close gaps. Because MUMmer is good at finding highly matched sequences very quickly, and GLASS is good at using a recursive algorithm to close gaps, which is more sensitive in finding conserved regions than local alignment. UniMarker method could also be included to combine its anchors with MUMs and GLASS anchors. A large portion of sets of anchors should be overlapping, but the marginal differences may have a great influence on the results sensitivity and specificity. (4) Hidden Markov Model for multiple whole genome alignment. More study is expected in the field of multiple genome alignment. As widely used in local alignment, HMM is good in finding profiles, thus could be used to search anchors and conserved domains among specie genomes. Each anchor could serve as a state in HMM, while an anchor itself could be a smaller HMM. However, using HMM will surely increase computation and time, and it could be useful when more sensitive global alignment is needed.

REFERENCES

1. Batzoglou, S. et al. 2000. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Research*. 10: 950-958
2. Brudno, M., Morgenstern B. 2002. Fast and sensitive alignment of large genomic sequences. *IEEE Computer Society Bioinformatics Conference*
3. Brudno, M. et al. 2004. Automated whole-genome multiple alignment of rat, mouse and human. *Genome Research*. 14: 685-692

4. Chen, L.Y.Y., Lu, S.H., Shih, E.S.C., Hwang, M.J., 2002. Single nucleotide polymorphism mapping using genome-wide unique sequences. *Genome Research*. 12: 1106-1111

5. Couronne, O., et al. 2003. Strategies and tools for whole-genome alignments. *Genome Research*. 13: 73-80

6. Delcher, A. et al. 1999. Alignment of whole genomes. *Nucleic Acids Research*. 27: 2369-2376

7. Deogun. J. et al. 2003. A prototype for multiple whole genome alignment. *36th Hawaii International Conference on System Sciences.*

8. Hohl, M., Kurtz, S., Ohlebusch, E. 2002. Efficient multiple genome alignment. *Bioinformatics*. 18: S312-S320

9. Liao, B.Y., Chang, Y.J., Ho, J.M., Hwang, M.J. 2004. The UniMarker (UM) method for synteny mapping of large genomes. *Bioinformatics*. 20: 3156-3165

10. Martins, W.S., Cuvillo, J., Cui, W., Gao, G.R. 2001. Whole genome alignment using a multithreaded parallel implementation. *Proceedings of the 13th Symposium on Computer Archetecture and High Performance Computing*.

11. Schwartz, S. et al. 2000. PipMaker – a web server for aligning two genomic DNA sequences. *Genome Research*. 10: 577-586

12. Schwartz, S. et al. 2003. Human-mouse alignments with BLASTZ. *Genome Research*. 13: 103-107