Ross Metzger
June 4, 2004
Biochemistry 218

Multiple Alignment of Genomic Sequences

Genomic sequence is currently available from ENTREZ for more than 40 eukaryotic and 157 prokaryotic organisms. As part of the ongoing NIH Intramural Sequencing Center's, Comparative Vertebrate Sequencing project, genomic sequences will soon be available from 50 vertebrates for regions orthologous to defined regions of the human genome. Managing and interpreting these sequence data requires new computational tools, including programs designed to align multiple genomic sequences. Biologists can use such alignments to identify functional elements (coding regions and transcription factor binding sites, as well as highly conserved elements whose exact function(s) remain to be determined, e.g. the recently described "ultraconserved" elements (Bejerano et al., 2004)), to understand the evolution of genome sequence and structure, and for phylogenetic analysis (for reviews see (Boffelli et al., 2004; Dubchak and Frazer, 2003; Frazer et al., 2003; Ureta-Vidal et al., 2003).

The goal of an alignment program is to align orthologous positions, i.e. positions in the sequences to be aligned that descend from the same position in the ancestral sequence. Programs should be as sensitive as possible, aligning as much orthologous sequence as possible, but should also be as precise as possible— only orthologous sequences should be aligned. (Non-orthologous sequences should either not be aligned, or matched to a gap.) Alignment programs can be used to align multiple whole genomes or to align multiple large genomic sequences.

Genomes evolve by rearrangements, inversions, and duplications, and contain repetitive elements, all of which can pose problems for alignment tools. Programs that use a global alignment strategy assume that orthologous regions are found in the same order in all the sequences to be aligned. For whole genomes, this assumption, then, is false. (Local alignment programs can detect transpositions, inversions, and duplications, but may do worse than global aligners at detecting orthologous regions in widely diverged sequences.) I will discuss five programs (MultiPipMaker, Multi-LAGAN, CHAOS-DIALIGN, MAVID, and TBA) designed to align multiple genomic sequences that produce local or global multiple alignments. Those programs that produce global multiple alignments (all except MultiPipMaker) assume that order within the orthologous sequences to be aligned is conserved. (Global alignment programs can be used to align whole genomes if the genomes are first broken down into chunks, e.g. by local aligners, in which conservation of order is assumed. See, for example, Brudno et al, 2004.) This may not always be the case, however, because small scale rearrangements can occur (Kent et al., 2003). Repetitive elements are dealt with either by removing them before aligning the sequences, or masking them initially, so that they are allowed to be aligned only if they are adjacent to aligned non-repeat regions. Both of these approaches require that species-specific repetitive sequences can be identified.

Programs for aligning multiple genomic sequences

MultiPipMaker

MultiPipMaker, available as a web-based server (http://bio.cse.psu.edu/pipmaker), generates true multiple alignments of long DNA sequences (Schwartz et al., 2003a). It returns all local alignments that score above a specified threshold. MultiPipMaker begins by generating a multiple alignment using local pairwise alignments between a reference sequence and each of the other sequences computed by the BLASTZ program (Schwartz et al., 2003b). This initial, crude multiple alignment is then refined to generate a true multiple alignment.

BLASTZ is a local alignment tool, which generates a set of local alignments using a Gapped BLAST-like strategy. BLASTZ finds short near-exact matches (sequences must match at 12 specific positions within runs of 19 nucleotides; a transition is allowed at any one of the 12 positions). These matches are then extended in both directions, not allowing gaps, until the score drops below some threshold. (The scores of low complexity sequence matches are downweighted.) Ungapped matches that score above a certain threshold are then extended using a dynamic programming method that allows for gaps. BLASTZ then searches in between each pair of adjacent alignments for 7-mer exact matches and allows a lower threshold to determine which ungapped matches to extend. (The idea is to use less strict criteria to align sequences in between those that align based on stricter criteria.) The local pairwise alignments are pruned to eliminate any overlaps, and then strung together. These strung together pairwise alignments, then, contain gaps within the local alignments (which are penalized using affine gap penalties) and gaps between local alignments. In constructing and refining the multiple alignment, gaps between these local alignments are not penalized.

The crude multiple alignment is assembled from these strung together pairwise alignments with the common reference sequence, and then refined using an iterative procedure designed to produce an optimal multiple alignment score. Each sequence within defined sub-regions (a sub-region in which there is no internal gap in that sequence) is removed from the alignment, the alignment adjusted to close any internal gap found in all the remaining sequences, and the removed sequence is realigned. MultiPipMaker uses the BLASTZ alignment scoring system to score nucleotide substitutions for all pairwise and multiple alignments. This matrix was optimized for human-mouse comparisons and so may not be optimal for comparisons of sequence from other organisms (Chiaromonte et al., 2002). (Most of the programs use the same scoring matrix and gap penalties for all organisms, though species-specific ones can be implemented (Brudno et al., 2004). As more analysis of available genomes is done, more realistic scoring tools, modeling gap distribution, e.g., can be developed (Kent et al., 2003).)

MultiPipMaker requires that only the reference sequence be finished quality. All other sequence can be provided as draft quality, in unordered or unoriented contigs. MultiPipMaker projects other sequences onto the reference sequence; which sequence is chose as the reference will affect the resulting alignment.

Multi-LAGAN

MLAGAN (Multi-Limited Area Global Alignment of Nucleotides), accessible as a web-based server (http://lagan.stanford.edu), uses a progressive alignment strategy to construct a multiple alignment, which can then be improved using an iterative refinement strategy (Brudno et al., 2003b). MLAGAN begins by creating a global pairwise alignment between the two evolutionarily closest sequences. This requires that the phylogenetic relationship among the sequences being aligned be known. MLAGAN uses a binary phylogenetic tree as a guide to determine what pairwise alignments to generate. To align sequences from human, chimpanzee, mouse, rat, and chicken, for example, MLAGAN first generates a human-chimpanzee pairwise alignment, then a mouse-rat alignment. This pair of pairwise alignments are then aligned, and the resulting alignment is then aligned to the chicken sequence.

MLAGAN generates the global pairwise alignments using LAGAN, which relies on anchoring to reduce computational time (Brudno et al., 2003b). Local similarities between the sequences are detected using the CHAOS algorithm, which looks for short, exact matches. A set of local similarities is ordered and strung together using a scoring matrix to form a rough global alignment. The ordered local alignments serve as anchors. The space searched by dynamic programming for an optimal alignment is limited to within a certain distance around the anchors.

MLAGAN uses a multiple alignment scoring system where matches and mismatches are scored as the sum of all pairwise combinations, though the scoring matrix is not discussed. Because MLAGAN uses a progressive alignment strategy, the initial pairwise alignments are fixed even as more sequences are added to the alignment. MLAGAN allows the option of refining the multiple alignment by removing each sequence and realigning it locally to the remaining sequences using anchors.

CHAOS-DIALIGN

DIALIGN is another global alignment program that can be used in combination with CHAOS to generate multiple alignments of genomic sequences (Brudno et al., 2003a). The CHAOS-DIALIGN combination can also be accessed over the web (http://dialign.gobics.de/chaos-dialign-submission). DIALIGN's ability to align long nucleotide sequences was limited by the program's long running time, since it required computational time proportional to the product of the sequences to be aligned. The CHAOS-DIALIGN combination uses CHAOS to find and extend local sequence similarities between all possible pairs of sequences to be aligned. Anchors are chosen from among the set of similarities to form a consistent set, starting with the highest scoring similarities. Using CHAOS in combination with DIALIGN significantly reduces computing time, but the CHAOS-DIALIGN combination may still be too slow to use to align multiple large genomic sequences (Bray and Pachter, 2004).

MAVID

MAVID, available as a web server (http://baboon.math.berkeley.edu/mavid), like MLAGAN, uses a progressive alignment strategy in which a binary phylogenetic tree determines which pairwise alignments to make (Bray and Pachter, 2004). Unlike MLAGAN, MAVID does not require that such a tree be supplied; it generates the guide tree. MAVID aligns alignments at internal notes of the tree by first inferring an ancestral sequence for each alignment, then aligning these ancestral sequences, and using this alignment to construct a multiple alignment. The pairwise alignments of the ancestral sequences are made using the AVID program. AVID is a global alignment program which like LAGAN uses anchors to start an alignment (Bray et al., 2003). AVID uses exon matches (determined using GENSCAN gene predictions and the translated BLAT tool for protein alignments) both as a set of pairwise anchors and as a set of constraints on the order of anchors in the multiple alignment. (This is similar to the way CHAOS-DIALIGN uses pairwise local similarities to constrain the ordered anchor set.) Regions between anchors are aligned using the Smith-Waterman algorithm with a scoring matrix and gap opening and extension penalties that depend on evolutionary distance.

TBA

MultiPipMaker generates an initial multiple sequence alignment by first making pairwise alignments of each of the sequences to be aligned with a common reference sequence. One limitation of this approach is that orthologous regions of a subset of sequences not present in the reference sequence may not be aligned. So the final multiple alignment may depend on which sequence is chosen as the reference. Analogously, a premise of the progressive alignment strategies used by MLAGAN and MAVID is that the order of pairwise alignments does matter. The multiple alignment refinement steps included in MultiPipMaker and MLAGAN are intended to compensate for this.

TBA (Threaded Blockset Aligner), not available as a web-based server, was designed to overcome such limitations of what Blanchette *et al.* refer to as “reference sequence” alignments (Blanchette et al., 2004). TBA aims to produce a set of local alignments (“blocks”) in which each position of each sequence to be aligned appears once and only once (a “threaded blockset”) and in which all significant alignments between some or all of the sequences are represented. (CHAOS-DIALIGN also tries to find significant pairwise local alignments between all sequences.) TBA produces a global multiple alignment by joining the blocks together. Blocks are generated using BLASTZ to produce pairwise alignments and a new program, MULTIZ, to align 3 or more sequences. Though in principle a threaded blockset can include duplications and inversions, the program currently cannot handle such a blockset, so it too can only align sequences in which order is conserved. The alignment produced by TBA can be displayed using any of the sequences as the reference by the Gmaj viewer. This allows the user to see conservation among sequences that might not be as easily apparent with a fixed reference.

Evaluating the tools

The papers presenting the multiple sequence alignments programs discussed above also include some kind of (formal or informal) evaluation and/or comparison with other alignment tools (Blanchette et al., 2004; Bray and Pachter, 2004; Brudno et al., 2003a; Brudno et al., 2003b; Schwartz et al., 2003a). Each uses a different evaluation method, and each program does best according to the method chosen.

Two kinds of strategies are used to test the ability of multiple sequence alignment programs to align orthologous positions: available biological data can serve as the standard to which the multiple alignment can be compared, and diverged sequences can be generated *in silico* for which the correct alignment— which positions derive from which in the original sequence— is known. Bray and Pachter and Brudno *et al.* use existing data to assess how well multiple alignment programs aligned exons in genomic sequence alignments, whereas Blanchette *et al.* use a simulation approach to assess how well the programs align neutral regions (sequence not under evolutionary selective pressure).

Bray and Pachter and Brudno *et al.* perform multiple alignments of genomic sequences orthologous to the ~1.8 Mb region on human chromosome 7 containing the *CFTR* gene (7q31) generated as part of the NIH Intramural Sequencing Center Comparative Sequencing Program (Thomas et al., 2003). As a standard for comparison, they use computational tools (Tblastx and Tblastn) to find the orthologues for all human exons in the region in each of the non-human sequences. Unfortunately, then, their evaluation depends on their ability to identify the orthologous exons. (Only a small number of coding sequences in this region from the organisms sequenced are currently available from RefSeq, I found. But even using RefSeq sequences still relies on alignment programs to determine orthologous sequences (Blanchette et al., 2004).) The ability to assess alignments using biological data is only as good as the available data. Assessing how well the programs align known orthologous non-coding sequence using biological data is more difficult given our lack of knowledge of orthologous sequences of other kinds.

To get around these limitations, Pollard *et al.* and Blanchette *et al.* create a set of diverged sequences whose alignment is known, using models of evolution (Blanchette et al., 2004; Pollard et al., 2004). This approach depends, not on the ability to identify orthologous sequences, but on the ability to generate them realistically. Accurate simulations require knowledge of kinds of evolutionary changes and their frequency. And this approach, then, may also depend on computational tools such as alignment programs to identify evolutionary changes. But a simulation approach allows Blanchette *et al.* to evaluate multiple alignments of neutrally evolving sequences, which could not be done using a biological data approach. The simulations Blanchette *et al.* use includes some features not used by Pollard *et al.*, who use their simulation to compare pairwise alignment tools: context dependent substitutions, empirically derived rates and sizes of insertions and deletions, and actual repetitive elements, but may also include biologically unrealistic assumptions such as a uniform rate of evolution across the sequence.

None of the papers discussed offers a comparison of all of the available programs for aligning multiple genomic sequences, nor do any of them attempt to formally evaluate how well the programs align evolutionarily constrained non-coding sequences such as transcriptional regulatory regions. Not only are different sets of programs compared using different benchmarks, but different measures of alignment quality are calculated in each case. Each of these differences may favor some algorithms over others in ways that may be hard to determine without systematic comparisons. (Programs that use anchors to constrain alignments may do better at aligning coding sequence at the expense of aligning neutrally evolving sequence.) Additionally, quality scoring systems should reflect differences between alignments that matter. For example, Blanchette *et al.* suggest that whether the sequence AA is aligned to –A or A- may not be important for some purposes such as identifying conserved regions or inferring ancestral sequence. A systematic comparison of the programs could be done on alignments of the sequences for regions orthologous to human 7q31 available from NISC for 22 vertebrates (http://www.nisc.nih.gov/open_page.html?/projects/comp_seq.html), using available RefSeq sequences and computational determinations of exons as a benchmark to compare coding sequence alignments, and using a simulation procedure similar to that of Blanchette *et al.* but which also allows for evolutionary constraints to be included (as Pollard *et al.* do) to compare alignment of both constrained and unconstrained sequence.

Brudno *et al.*, Blanchette *et al.*, and Bray and Pachter evaluate their programs by extracting and scoring selected pairwise alignments from the multiple alignment. They compare only these pairwise alignment scores, and do not attempt to calculate a score for the multiple alignment, for example, by calculating and summing all possible pairwise scores (Lassmann and Sonnhammer, 2002).

One of the reasons to do multiple alignments is to get more information than pairwise alignments alone can yield, so another criteria for evaluating multiple alignment programs might be, as Blanchette *et al.* suggest, that alignments should improve as the number of species aligned increases. But there may have to be trade-offs. If there is information to be gotten from an alignment of a large number of species than cannot be gotten from alignments of smaller numbers of species, it may be that larger multiple alignments that score somewhat worse when compared by extracting pairwise alignments are still desirable. So we may need to come up with ways of comparing multiple alignments that will allow us to determine how much information we can get out of them.

Different multiple alignment programs may work better for some kinds of genomic sequence (coding, functional non-coding, or neutrally evolving) and for different purposes. Even after evaluating the sensitivity and precision of multiple alignment programs to measure how well they align orthologous positions, there is still the challenge of how to evaluate alignments in terms of their usefulness. We want to get as accurate a multiple alignment as possible because we want to be able to use the alignments to discover new regions and features of genomes to evaluate functionally, and to understand their evolution. More than one alignment, however, will correspond to the same accuracy score. We need to find ways of discriminating among them, of determining experimentally what in an alignment is biologically significant and insignificant.

References

Bejerano, G., Pheasant, M., Makunin, I., Stephen, S., Kent, W. J., Mattick, J. S., and Haussler, D. (2004). Ultraconserved elements in the human genome. Science *304*, 1321-1325.

Blanchette, M., Kent, W. J., Riemer, C., Elnitski, L., Smit, A. F., Roskin, K. M., Baertsch, R., Rosenbloom, K., Clawson, H., Green, E. D., *et al.* (2004). Aligning multiple genomic sequences with the threaded blockset aligner. Genome Res *14*, 708-715.

Boffelli, D., Nobrega, M. A., and Rubin, E. M. (2004). Comparative genomics at the vertebrate extremes. Nat Rev Genet *5*, 456-465.

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

Bray, N., and Pachter, L. (2004). MAVID: constrained ancestral alignment of multiple sequences. Genome Res *14*, 693-699.

Brudno, M., Chapman, M., Gottgens, B., Batzoglou, S., and Morgenstern, B. (2003a). Fast and sensitive multiple alignment of large genomic sequences. BMC Bioinformatics *4*, 66.

Brudno, M., Do, C. B., Cooper, G. M., Kim, M. F., Davydov, E., Green, E. D., Sidow, A., and Batzoglou, S. (2003b). LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. Genome Res *13*, 721-731.

Brudno, M., Poliakov, A., Salamov, A., Cooper, G. M., Sidow, A., Rubin, E. M., Solovyev, V., Batzoglou, S., and Dubchak, I. (2004). Automated whole-genome multiple alignment of rat, mouse, and human. Genome Res *14*, 685-692.

Chiaromonte, F., Yap, V. B., and Miller, W. (2002). Scoring pairwise genomic sequence alignments. Pac Symp Biocomput, 115-126.

Dubchak, I., and Frazer, K. (2003). Multi-species sequence comparison: the next frontier in genome annotation. Genome Biol *4*, 122.

Frazer, K. A., Elnitski, L., Church, D. M., Dubchak, I., and Hardison, R. C. (2003). Cross-species sequence comparisons: a review of methods and available resources. Genome Res *13*, 1-12.

Kent, W. J., Baertsch, R., Hinrichs, A., Miller, W., and Haussler, D. (2003). Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. Proc Natl Acad Sci U S A *100*, 11484-11489.

Lassmann, T., and Sonnhammer, E. L. (2002). Quality assessment of multiple alignment programs. FEBS Lett *529*, 126-130.

Pollard, D. A., Bergman, C. M., Stoye, J., Celniker, S. E., and Eisen, M. B. (2004). Benchmarking tools for the alignment of functional noncoding DNA. BMC Bioinformatics *5*, 6.

Schwartz, S., Elnitski, L., Li, M., Weirauch, M., Riemer, C., Smit, A., Green, E. D., Hardison, R. C., and Miller, W. (2003a). MultiPipMaker and supporting tools: Alignments and analysis of multiple genomic DNA sequences. Nucleic Acids Res *31*, 3518-3524.

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

Thomas, J. W., Touchman, J. W., Blakesley, R. W., Bouffard, G. G., Beckstrom-Sternberg, S. M., Margulies, E. H., Blanchette, M., Siepel, A. C., Thomas, P. J., McDowell, J. C*., et al.* (2003). Comparative analyses of multi-species sequences from targeted genomic regions. Nature *424*, 788-793.

Ureta-Vidal, A., Ettwiller, L., and Birney, E. (2003). Comparative genomics: genome-wide analysis in metazoan eukaryotes. Nat Rev Genet *4*, 251-262.