Handling Rearrangements in DNA Sequence Alignment

Maneesh Bhand

12/5/10

1 Introduction

Sequence alignment is one of the core problems of bioinformatics, with a broad range of applications such as genome assembly, gene identification, and phylogenetic analysis [1]. Alignments between DNA sequences are used to infer evolutionary or functional relationships between genes. Evolution occurs through DNA mutations, which include small-scale edits and larger-scale rearrangement events. Traditional sequence alignment algorithms, such as the Needleman-Wunsch global alignment [2] and Smith-Waterman local alignment [3] algorithms, and their heuristic-based successors, are not capable of accounting for rearrangements in their scoring metric, and are not suitable for inferring the evolutionary history of aligned sequences. With the advent of whole genome sequencing and the exponential increase in the amount of available sequence data, the need for algorithms capable of inferring homologies across entire genomes has become more acute. In the past decade, several alignment algorithms have been developed to account for chromosomal rearrangements. This paper provides a critical overview of these algorithms, with a focus on their treatment of rearrangements. The advantages and limitations of these algorithms are explored, along with a discussion of potential future developments in the problem of handling rearrangements.

2 Mutations in the Genome

The process of evolution is shaped by DNA mutations, which occur primarily during the process of replication. The most common mutations are small-scale substitutions, insertions, and deletions, which involve one or several base pairs. Larger-scale rearrangements of genomic subsequences also occur.

Historically, sequence alignment algorithms have focused on the small-scale events, by maximizing scoring functions that include bonuses for aligning nucleotides and penalties for substitutions and gaps. Global and local alignment methods are therefore well-suited for analyzing small-scale mutations and aligning regions containing only these types of mutations. Very large mutations are also readily identifiable; large-scale rearrangements of greater than 1 megabase in length can be detected by chromosomal mapping techniques such as fluorescence in situ hybridization (FISH) [4]. It is the intermediate class of local rearrangements, ranging from a few hundred to a few hundred thousand bases, that is of particular interest to sequence alignment [17,18], and for which the question of how best to compute and evaluate alignments remains the most open.

Mutations in the genome can be characterized as follows:

- Point mutations (or substitutions) are mutations in which one base pair is substituted for another. These are the most common mutation events, and are responsible for the single nucleotide polymorphisms (SNPs) that have been studied extensively in the literature.
- Insertions are mutations in which novel DNA sequence is added to the sequence. Insertions typically involve a few base pairs; larger insertions are due to duplication events.
- Deletions are mutations in which a section of DNA is removed from the sequence. The length of the removed sequence can range from a few bases to many megabases.
- Duplications are mutations in which a section of DNA is copied and inserted elsewhere in the DNA. They are critical for the development of paralogous genes.
- Inversions are mutations in which a section of DNA is removed from the sequence and re-inserted in the same location, but in the opposite orientation.
- Translocations are mutations in which a section of DNA is removed from the sequence and inserted in a different location in the same orientation.

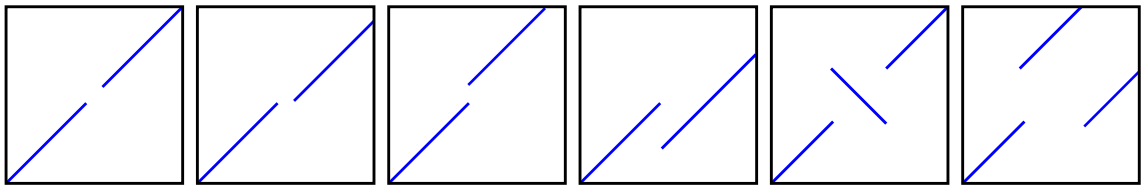

Figure 1: Diagram of rearrangements, from left to right: point mutation, insertion, deletion, duplication, inversion, and translocation. Mutations are in the first sequence (horizontal direction). Blue lines indicate areas of homology with the second sequence (vertical direction).

A few other mutation events exist, but they are either on too large a scale to be relevant to sequence alignment (e.g. nondisjunction events), or they can be expressed as a combination of the above mutations (e.g. a translocated inversion). Therefore, an algorithm that is able to accurately model these six mutation types will be able to infer likely homology and align sequenced genomes. It should be noted that these mutations are not evenly distributed throughout the genome [17]; certain areas (about 5% of the genome) comprise mutation "hotspots" and show higher frequencies of rearrangement. Conversely, functional regions that are highly conserved will show much fewer rearrangements due to negative selective pressure.

3 Sequence Alignment

An alignment between two sequences is a mapping from the nucleotides of one sequence to those of another. An alignment can be thought to represent one of two ideas: a mapping based on the evolutionary history of the two sequences, in which aligning bases are derived from some inferred ancestral

sequence, or a mapping based on functional relationships between genomic regions, in which aligned bases have common functionality in the organism [1]. In the case of whole genome alignment, in which rearrangements are present, the goal of alignment is typically to detect homologous regions based on common ancestry, so this review will consider alignments as such.

3.1 Early Approaches

This section provides a brief overview of early approaches to alignment and homology detection; these methods are not, by themselves, currently used to align sequences containing rearrangements, but they are often incorporated into more complex algorithms that are.

3.1.1 Global Alignment

The earliest approach to alignment is global alignment, which seeks to find the optimal transformation from one sequence to another by some combination of nucleotide substitutions, insertions, and deletions. The quality of an alignment is determined by a scoring function, which assigns a score to every aligning pair of nucleotides, and a penalty for gaps in the alignment.

Global alignment requires that alignments be increasing in both strands; in other words, if the strands are laid on top of each other and aligning bases are connected with lines, then the lines cannot cross. This means that there is no way of recognizing duplications, inversions, or translocations; a duplication in one strand would be interpreted as deletions in the other, an inversion would be scored as a sequence of substitutions, and a translocation would appear as a deletion in each strand.

The Needleman-Wunsch algorithm [2] uses dynamic programming to compute the optimal alignment in polynomial time; this is not computationally feasible for long sequences, so heuristic-based alternatives, which sacrifice optimality in order to gain speed, are more commonly used. Some popular global alignment algorithms include Dialign [5], Avid [6], and LAGAN [7]. Because global alignment is capable of handling the nucleotide-scale mutations (point mutations, insertions, and small deletions), it is often used to align smaller sequences considered to be orthologous by later algorithms that handle rearrangements.

3.1.2 Local Alignment

Local alignment is an extension of global alignment that removes the constraint that the mapping must occur for the entire length of the strands; instead, local alignment will detect subsequences within the input sequences that align to each other. This allows for detection of rearrangements, as local alignment will return a set of hits corresponding to homologous regions within the aligned sequences.

However, local alignment cannot explain how the two sequences evolved from a common ancestral sequence [1]. Furthermore, the hits returned by local alignment may be subsequences of larger syntenic (without rearrangements) blocks, which need to be detected to infer homology. Also, since the scoring system is unchanged from the original Needleman-Wunsch formulation, there is no inherent capacity for rearrangements, so the score of an alignment is not always proportional to its evolutionary likelihood. It is often difficult to set the score threshold for significance, as high thresholds will miss homologous regions and low thresholds will return spurious similarities. Nevertheless, local alignment is a critical component in the algorithms used to handle rearrangements, which assemble locally aligned regions into larger alignments.

The Smith-Waterman algorithm [3] extends Needleman-Wunsch to compute optimal local alignments. As with global alignment, the dynamic programming approach is not tractable for large sequences, so heuristic-based aligners, which typically speed up alignment by searching for matching k-mers within the strings, are used in practice. Some popular local alignment algorithms include BLASTZ (and its successor LASTZ) [8], PatternHunter [9], and CHAOS [10].

3.1.3 Extensions of the Scoring Scheme

There have been a few efforts to extend the scoring scheme of the Needleman-Wunsch algorithm to include rearrangement events:

- The DSI scoring model [11] extends the scoring system of local alignment in order to handle tandem duplications (duplications in which the two copies are next to each other); with this extension, the running time of the dynamic programming solution increases to $O(n^4)$, so heuristic algorithms must be used. While this model allows for duplication events, there is no capacity for inversions, translocations, or non-tandem duplications. Consequently, this algorithm has seen little use.

- Another extension of the Smith-Waterman algorithm has been made to account for non-overlapping inversions [12]. This approach includes a fixed cost for adding an inversion and allows for separate match matrices and gap penalties for the inverted and non-inverted cases. As with the previous algorithm, finding the optimal solution is not tractable, and so heuristic-based methods (evaluating only a set of the most likely inversions in this case) are needed. Because the algorithm allows for different parameters for inverted strings, there is some flexibility in tuning the algorithm's performance to detect inversions. However, the algorithm cannot represent overlapping inversions or other rearrangements.

3.1.4 Limitations in Homology Inference

Even if we restrict ourselves to regions that do not contain rearrangements, there is uncertainty in inferring homologies between the sequences, which will lead to alignment errors. Lunter et al. [16] identifies three sources of uncertainty in alignments: The first is "gap wander", in which natural mutations introduce spurious local similarities between sequences, which compete with and cannot be distinguished from actual homologies, causing gaps to be assigned to incorrect positions. The second is "gap attraction", when two indels are very close to each other, and the alignment algorithm prefers one larger gap instead of two smaller gaps (due to fixed penalties for opening gaps). The third is "gap annihilation", in which both sequences have a gap of the same size (i.e. an indel in each sequence), but the algorithm aligns the sequences without gaps, avoiding the gap penalty. It is important to note these issues with local alignment, because later algorithms will build upon these alignment blocks in producing homology maps.

3.1.5 Synteny Mapping Algorithms

One of the drawbacks of local alignment is its inability to explain ancestral relationships of aligned sequences. To rectify this, a new class of mapping algorithms was developed, based on the idea of constructing a set of local alignments and then grouping together alignment blocks to find regions of synteny (orthologous regions that can be converted to each other by a sequence of rearrangements) [1]. Some examples of this approach include:

- Waterston et al. [13] used PatternHunter to generate local alignments greater than some size. These alignments, called anchors, were grouped into syntenic segments when they occurred on the same chromosome and in the same orientation. These were then grouped into larger syntenic blocks [1].

- GRIMM [14] arranged its anchor segments in a graph, with edges connecting anchors whose distance was smaller than some pre-specified threshold. The connected components of this graph were identified as syntenic regions [1].

- Couronne et al. [15] combined local and global alignment by performing a local alignment using BLAT, taking the top-scoring hits, consolidating nearby hits into regions, and performing global alignment on these regions.

The principal disadvantage to these approaches is that their resolution of rearrangements is insufficient; these algorithms are able to detect larger-scale rearrangements, but smaller (e.g. a few hundred bases) rearrangements are either contained in local alignment blocks (and therefore not scored as rearrangements), or not aligned in the final output. The concept of building a homology map by combining local alignment blocks, however, is a very useful one, and further refinements to this approach have been more successful in identifying rearrangements.

3.2 Chains and Nets

Perhaps the most commonly used system for handling rearrangements is the chains and nets of the UCSC genome browser [17]. The critical insight behind chains/nets is that local alignment blocks can be stitched together in order to form a global mapping, and that assembling "chains" of local alignments will result in a mapping that allows for local rearrangements.

In brief, the algorithm works by using BLASTZ to construct local alignment blocks. These blocks are assembled into structures called chains, which are then assembled into a net, which defines the mapping between homologies in the two species. In this way, the nucleotide-level mutations are handled by BLASTZ, and the rearrangements are handled by the chaining and netting procedure.

A chain is defined as a collection of local alignment blocks, such that none of the blocks are overlapping, all of the blocks are in the same direction, and the ordering of blocks within the chain is consistent with the ordering of the genome sequence of both species. Therefore, a chain might represent non-contiguous sections of DNA that could have been derived from a common ancestral sequence without any rearrangement.

The program for constructing chains, called AXTCHAIN, uses a k-dimensional tree to efficiently compute maximal chains - that is, by adding alignment blocks to the longest subchains they are consistent with. Scoring parameters are included for gap penalties in between the local alignment blocks.

Once the chains have been computed, they are assembled into a net using a greedy procedure. First, the list of chains is sorted by score and the highest scoring chain is added to the net. Then, each remaining chain (in order of decreasing score) is used to try to fill in the gaps of the net. Low-scoring chains that are made redundant by higher-scoring chains are thrown out, duplications (multiple chains covering the same bases) are noted, and the net is returned.

Chains and nets are a powerful tool for alignment, and have several advantages over other algorithms. First, the greedy assembly of the net has the advantages of being fast (as it is heuristic-based) and also of being able to filter out many of the spurious local alignment blocks detected by BLASTZ. Since the

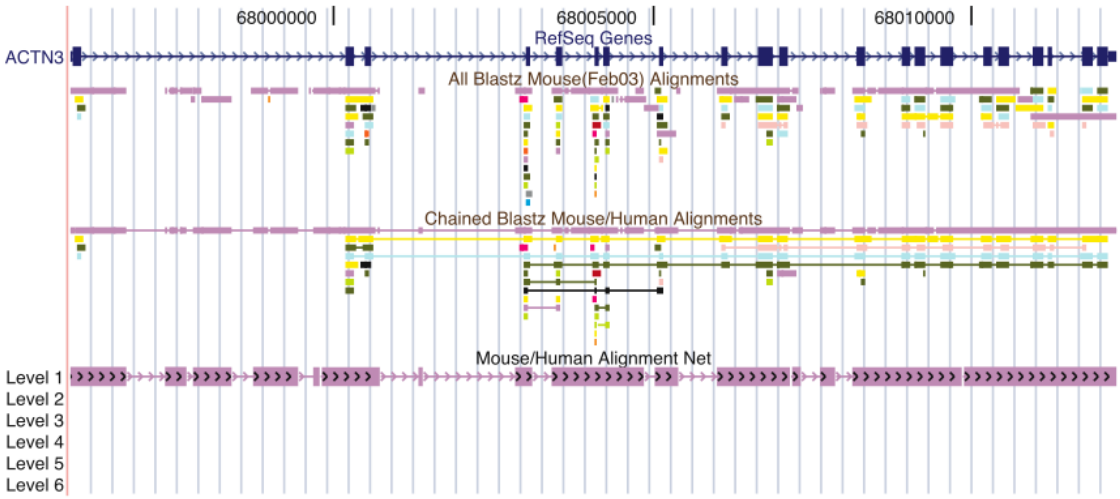

Figure 2: Mouse/human alignments as shown in the UCSC Genome Browser [17]. Local alignment blocks from BLASTZ (top) are organized into chains (middle), which are assembled into a net (bottom). Note that this process filters out much of the noise produced by local alignment, and returns a simpler mapping.

score of a chain is dependent on the scores of its constituent blocks, the algorithm implicitly favors a parsimonious explanation of ancestry inference when constructing the net, as larger chains that are able to explain more of the alignment are given priority.

One advantage of including gap penalties in scoring chains is that the algorithm now has two sets of gap penalties: the standard affine gap penalties that apply within local alignment blocks, and the new gap penalties for gaps within chains. This is more biologically meaningful formulation of gap penalties, as a linear penalty function is not appropriate for large gap sizes (very large gaps are not substantially worse than large gaps). That being said, there is also room for improvement, as more precise models of recombination would allow for more realistic gap penalties.

Chains and nets also have some drawbacks. First, the alignment of two genomes is not symmetric (i.e. the optimal alignment of A with B is not the same as the optimal alignment of B with A), because the nets are constructed with respect to one of the genomes, with chains from the other genome filling in gaps in the alignment. Secondly, there is no mathematical basis for the scoring of chains; unlike the Needleman-Wunsch scoring system, in which the match matrices can be given a probabilistic interpretation (so the score of an alignment is roughly proportional to the probability of its occurrence in nature), the scoring of chains has no such interpretation (although it could be argued that the algorithm is implicitly constructing the most likely homology mapping, based on the parsimony argument presented above). There are situations which are not handled ideally (for example, penalizing inversions in nets [17]), and some assumptions made by the algorithm which are sometimes false - that rearrangements are always independent and non-overlapping, and that translocations occur in the middle of chromosomes [17].

3.3 Shuffle LAGAN

A contemporaneous solution to handling alignments with rearrangements was presented in Shuffle-LAGAN (SLAGAN) [18], an extension of the LAGAN alignment algorithm. As with the chains and nets

of the UCSC Genome Browser, SLAGAN is based on chaining together local alignments to produce a global map.

Briefly, the algorithm first discovers local alignments between two sequence using the CHAOS tool, which works by chaining together short pairs of sequences calls seeds. Next, the algorithm constructs what is termed a *1-monotonic conservation map*, which is a chain of local alignments that do not overlap in the first sequence and has the maximum score over all possible such chains. There is some processing of the ends of these chains (because homologies at the ends may not be detected), and the resulting output is what the authors call a "glocal" alignment, a global alignment incorporating rearrangements.

One important advantage to Shuffle-LAGAN's scoring approach is that it allows for different penalties depending on the orientation of the strands in each local alignment. The algorithm's ability to handle rearrangements comes from chaining two local alignments L_1 and L_2 together, as either of the strands can be in the positive direction or negative direction (reverse complement), and either L_2 follows L_1 (in order) or L_1 follows L_2 (reverse order). Therefore, there are 8 options, which are assigned 4 different gap penalties: the regular gap penalties (2 positive strands chained in order or 2 negative chained in reverse order), inversions (a positive strand chained with a negative strand), translocations (positive and negative chained in order or negative and positive chained in reverse), or translocated inversions (positive and negative chained in reverse or negative and positive chained in order).

SLAGAN shares many of the limitations of chains and nets. As with chains/nets, SLAGAN produces alignments which are not symmetric, because of the constraint that alignments in chains should not be overlapping in the first sequence. This is necessary in order to make the problem of assembling the maximal chain tractable ($O(n^2)$), as the algorithm would be of exponential complexity otherwise. In addition, there is no solid mathematical basis for the scoring in the algorithm; the rearrangement penalties are arbitrary and chosen to produce reasonable synteny maps, but do not correspond to any probabilistic notion of evolutionary distance. A third issue is that, due to heuristic local aligners, the glocal alignment is not optimal; furthermore, when rearrangements overlap, correctly inferring the evolutionary history of a locus is not always possible, so the appropriate scoring functions may not be applied.

4 Multiple Sequence Alignment

One of the issues with aligning sequences based on their inferred evolutionary history is that there is not enough information contained in the genomes to allow for certain inference of the ancestral sequence; for example, a sequence containing two inverted segments near each other could be the result of two non-overlapping inversions, or the result of a smaller inversion inside a larger inversion. In fact, there are an infinite number of possible evolutionary histories for any two sequences, so alignment algorithms must try to find the most plausible history in constructing alignments. One strategy for improving this inference is to align many genomes at once. Multiple sequence alignment allows for algorithms to take advantage of phylogenetic relationships between organisms to infer putative ancestral sequences and align homologous sequences.

There are a multitude of multiple sequence alignment algorithms available, including several of the aligners mentioned previously, which have been extended to work for multiple sequences. However, these algorithms have largely focused on local alignments, or alignments at higher scales (of genes or amino acids). One recent MSA algorithm that handles rearrangements is the SuperMap algorithm, an extension of SLAGAN based on progressive alignment of sequences [19].

One of the limitations of SLAGAN was its asymmetry; SuperMap fixes this by running SLAGAN twice to generate two 1-monotonic maps, and merging the two maps together. The other extension is to align multiple sequences; SuperMap uses a progressive alignment framework, in which the most closely related (based on phylogenetic trees) organisms are aligned first, and then these alignments are aligned with other alignments in order to build up the final alignment. Although SuperMap is still subject to many of the same limitations that SLAGAN has, it is a promising development because of its ability to incorporate information from multiple sequences in inferring homologies.

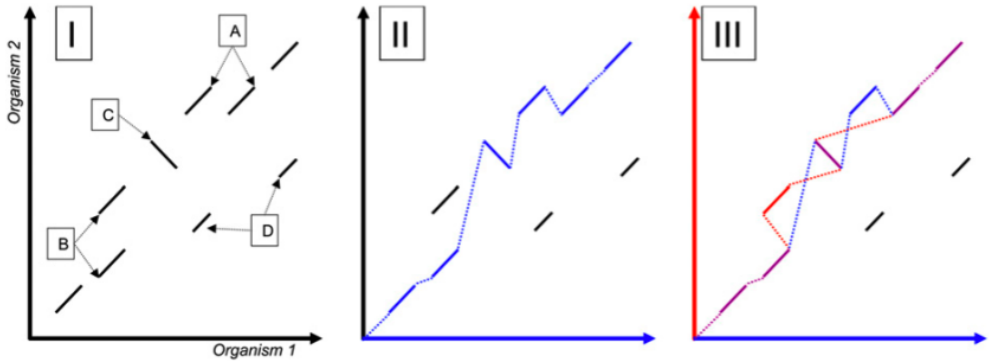

Figure 3: Diagram of the SuperMap algorithm [19]. The algorithm constructs 1-monotonic maps for both organisms. The regions where these maps overlap are syntenic blocks.

5 Future Directions

Sequence alignment remains an open problem, particularly with regard to local rearrangements. Current approaches to handling rearrangements largely revolve around the assembly of locally aligned blocks into homology maps. Yet, these approaches must make concessions to optimality and mathematical grounding for the sake of practicality. How might the future developments in sequence alignment unfold in the next decade?

The advent of high-throughput sequencing technologies has made the assembly of genomes more accessible, and we can expect more and more genomic sequence data to be available in the future. This will facilitate the development of mathematical models of DNA evolution, as we will be able to obtain much better estimates of the frequency of rearrangement events as a result of having more data available.

Furthermore, as more of the functionality of the genome is discovered, alignment algorithms will be able to take advantage of conservation in these alignment models, in order to help account for selective pressures in shaping genome evolution. At the same time, the development of improved alignment models will allow for better understanding of the functional regions of the genome, so there is some degree of bootstrapping that will be enabled by the advent of more genome assemblies.

Improved alignment algorithms, bolstered by newly available genomic sequences, will allow for a better understanding of the role of mutation events in evolutionary processes.

Sources:

1. S. Batzoglou
The many faces of sequence alignment.
Briefings in Bioinformatics 1: 6 22, 2005.

2. S.B. Needleman, C.D. Wunsch
A general method applicable to the search for similarities in the amino acid sequence of two proteins
J. Mol. Biol., Vol. 48, pp. 443-453

3. T.F. Smith, M.S. Waterman
Identification of common molecular subsequences
J. Mol. Biol., Vol. 147, pp. 195-197

4. D. Pinkel, J. Landegent, C. collins, J. Fuscoe, R. Segraves, J. Lucas, J. Gray
Fluorescence in situ hybridization with human chromosome-specific libraries: detection of trisomy 21 and translocations of chromosome 4
PNAS 85 (23) (1988) pp.9138-9142

5. B. Morgenstern
DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment
Bioinformatics, 15, pp.211-218

6. N. Bray, I. Dubchak, L. Pachter
AVID: a global alignment program
Genome Research, 13, pp.97-102

7. M. Brudno, C.B. Do, G.M. Cooper, M.F. Kim, et al.
LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA
Genome Research, 13, pp.721-731

8. S. Schwartz, W.J. Kent, A. Smit, Z. Zhang, R. Baertsch, R.C. Hardison, D. Haussler, W. Miller
Human-mouse alignments with BLASTZ
Genome Research, 13, pp. 103-107

9. B. Ma, J. Tromp, M. Li
PatternHunter: faster and more sensitive homology search
Bioinformatics 18, pp. 440-445

10. M. Brudno, B. Morgenstern
Fast and sensitive alignment of large genomic sequences
Proceedings of the Bioninformatics Conference (CSB), IEEE Computer Society, pp. 138-147

11. G. Benson
Sequence alignment with tandem duplication
Journal of Computational Biology 4 (3) (1997), pp. 351-367

12. M. Schoniger, M.S. Waterman
A local algorithm for DNA sequence alignment with inversions
Bulletin of Mathematical Biology 54 (4) (1992) pp. 521-536

13. R.H. Waterston, K. Lindblad-Toh, E. Birney, et al.
Initial sequencing and comparative analysis of the mouse genome
Nature 420, pp. 520-562

14. P.A. Pevzner, G. Tesler
Genome rearrangements in mammalian evolution: Lessons from human and mouse genomic sequences
Genome research (13), pp. 73-80

15. O. Couronne, A. Poliakov, N. Bray, T. Ishkhanov, D. Ryaboy, E. Rubin, L. Pachter, I. Dubchak
Strategies and tools for whole-genome alignments
Genome Research 13 (2003), pp. 73-80

16. G. Lunter, A. Rocco, N. Mimouni, A. Heger, A. Caldeira, J. Hein
Uncertainty in homology inferences: Assessing and improving genomic sequence alignment
Genome Research 18 (2008), pp. 298-309

17. W.J. Kent, R. Baertsch, A. Hinrichs, W. Miller, D. Haussler
Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes
PNAS 100(20) (2003) pp. 11484-9.

18. M. Brudno, S. Malde, A. Poliakov, C. Do, O. Couronne, I. Dubchak, S. Batzoglou
Glocal alignment: finding rearrangements during alignment
Bioinformatics Vol. 19 Suppl. 1 (2003)

19. I. Dubchak, A. Poliakov, A. Kislyuk, M. Brudno
Multiple whole-genome alignments without a reference organism
Genome Research. 19 (2009) pp. 682-689