Comparing Assemblies Using Fragments and Mate-Pairs

Daniel H. Huson, Aaron L. Halpern, Zhongwu Lai, Eugene W. Myers,
Knut Reinert, and Granger G. Sutton

Informatics Research, Celera Genomics Corp.
45 W Gude Drive, Rockville, MD 20850, USA
Phone: +1-240-453-3356, Fax: +1-240-453-3324
Daniel.Huson@Celera.com

Abstract. Using current technology, large consecutive stretches of DNA (such as whole chromosomes) are usually assembled from short fragments obtained by shotgun sequencing, or from fragments and mate-pairs, if a "double-barreled" shotgun strategy is employed. The positioning of the fragments (and mate-pairs, if available) in an assembled sequence can be used to evaluate the quality of the assembly and also to compare two different assemblies of the same chromosome, even if they are obtained from two different sequencing projects. This paper describes some simple and fast methods of this type that were developed to evaluate and compare different assemblies of the human genome. Additional applications are in "feature-tracking" from one version of an assembly to the next, comparisons of different chromosomes within the same genome and comparisons between similar chromosomes from different species.

1 Introduction

Although current technology for DNA sequencing is highly automated and can determine large numbers of base pairs very quickly, only about (on average) 550 *consecutive* base pairs (bp) can be reliably determined in a single read [6]. Thus, a large consecutive stretch of source DNA can only be determined by "assembling" it from short fragments obtained using a *shotgun sequencing* strategy [5]. In a modification of this approach called *double-barreled* shotgun sequencing [1], larger clones of DNA are sequenced from both ends, thus producing *mate-pairs* of sequenced fragments with known relative orientation and approximate separation (typically, employing a mixture of 2kb, 5kb, 10kb, 50kb and 150kb clones). So, usually a sequencing project produces a collection of fragments that are randomly sampled from the source sequence. The average number x of fragments that cover any given position in the source sequence is known as the *fragment x-coverage*.

Given two different assemblies of the same chromosome-sized source sequence, possibly obtained from two different sequencing projects, how can one evaluate and compare them? The aim of this paper is to present some fast and simple methods addressing this problem that are based on fragment and mate-pair data obtained in a sequencing project for the source sequence. Additional applications are in tracking

O. Gascuel and B.M.E. Moret (Eds.): WABI 2001, LNCS 2149, pp. 294–306, 2001.

forward "features" from one version of an assembly to the next, comparison of different chromosomes from the same genome and of similar chromosomes from different species. Although each method on its own is just an implementation of a simple idea or heuristic, our experience is that the integration of these methods gives rise to a powerful tool. We originally developed this tool to compare different assemblies of the human genome, see Figures 6 and 7 in [7].

In Section 2 we discuss assembly evaluation and comparison techniques based on fragments. In particular, we introduce the concept of "segment discrepancy" that measures by how much the positioning of a segment of conserved sequence differs between two assemblies. Then we present some mate-pair based methods in Section 3, including a useful breakpoint detection heuristic. Finally, we demonstrate the utility of these methods in Section 4.

2 Fragment-Based Analysis and Comparison Methods

Several useful methods for evaluating a single assembly or comparing two assemblies—such as sequencing coverage, dot-plots, or line-plots—can be implemented in terms of the positions in an assembly to which fragments are assigned.

For our purposes, a *contig* is simply a finite string $A = a_1 a_2 \ldots$ of characters $a_i \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}, \mathsf{N}\}$ representing a stretch of contiguous DNA, where A, C, G and T correspond to the four bases and N stands for "unknown". An *assembly* is a contig A that was obtained from the fragments of some sequencing project using some assembly algorithm, without elaborating on the details. A run of consecutive N's represents an undetermined sequence part, and the number of N's in the run is sometimes used to represent its estimated length.

A fragment is a string $F = f_1 f_2 \ldots$ of characters $f_i \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$, of length $\mathrm{len}(F)$ usually less than 900. We say that a fragment F *hits* (or is *recruited by*) an assembly A if F globally aligns to A with high identity (e.g. 94% or more). In this case, we use $s(F, A)$ and $t(F, A)$ to denote the position in A to which the first character and last character of F align to, respectively. In particular, a fragment aligns in the forward direction if $s(F, A) < t(F, A)$, whereas the alignment is against the reverse-complement of F if $s(F, A) > t(F, A)$. For simplicity, we will assume that all s values are distinct, i.e., $s(F) \neq s(G)$ for any two different fragments that hit A. (In practice, fragment coordinates do sometimes agree, but our experience is that one can simply ignore such fragments without a substantial loss of coverage.)

Given a set of fragments $\mathcal{F}$ and an assembly A, we use $\mathcal{F}(A)$ to denote the set of all fragments in $\mathcal{F}$ that hit A. If an assembly A was obtained by assembling fragments from a set $\mathcal{F}$, then the set $\mathcal{F}(A)$, and the values of $s(F, A)$ and $t(F, A)$ for all $F \in \mathcal{F}(A)$, are known. If an assembly A of a chromosome is obtained from one sequencing project, and the set of fragments $\mathcal{F}$ available was obtained from a different sequencing project studying the same chromosome, then a fast high-fidelity alignment program [4] can be used to compute $\mathcal{F}(A)$.

2.1 Fragment-Coverage Plot

Let A be an assembly and $\mathcal{F}(A)$ a set of fragments that hit A. For each fragment $F \in \mathcal{F}(A)$ define a *begin-event* $(\min\{s(F,A), t(F,A)\}, +1)$ and an *end-event* $(\max\{s(F,A), t(F,A)\}, -1)$. To obtain a *fragment-coverage plot* for A, consider all events (x, e) in order of their first coordinate x and for each begin-event, plot the number of fragments that span x, given by the number of begin-events minus the number of end-events seen so far, see Figure 1.

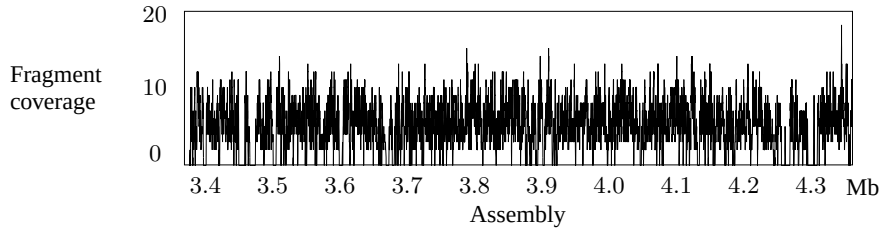

Fig. 1. Fragment-Coverage Plot for a 1 Mb Region of Chromosome 2 of Human [7]. The assembly A is represented by a line segment $[1, \text{len}(A)]$ along the x-axis. The number of fragments uniquely hitting A is plotted as a function of their position.

A fragment-coverage plot is useful because poorly assembled regions often have low fragment-coverage, whereas regions of repetitive sequence can be identified as those stretches of sequence that are hit by unusually high numbers of fragments.

In practice, one can easily accomodate for fragments hitting multiple times. However, for ease of exposition, throughout this paper we will assume that $\mathcal{F}(A)$ is the set of all fragments that *uniquely* hit A.

2.2 Dot-Plot and Line-Plot

Consider two different assemblies A and B of the same chromosome, and assume that a set $\mathcal{F}$ of fragments obtained from a shotgun sequencing project for the chromosome is given. Once we have determined $\mathcal{F}(A)$ and $\mathcal{F}(B)$, how can we visualize this data?

Let $\mathcal{F}(A,B) := \mathcal{F}(A) \cap \mathcal{F}(B)$ denote the set of fragments that hit both assemblies. A simple dot-plot can be produced by plotting (x, y) with $x := s(F,A)$ and $y := s(F,B)$ for all $F \in \mathcal{F}(A,B)$, see Figure 2; at higher resolution, plot a line from $(s(F,A), s(G,B))$ to $(t(F,A), t(G,B))$. Alternatively, represent assembly A and B by a line segment from $(1,0)$ to $(\text{len}(A), 0)$ and from $(1,1)$ to $(\text{len}(B), 1)$, respectively. A simple line-plot showing matching regions of the two assemblies is obtained by drawing a line segment between $(s(F,A), 0)$ and $(s(F,B), 1)$ for all $F \in \mathcal{F}(A,B)$, see Figure 3.

If $\mathcal{F}(A)$ is given, but $\mathcal{F}(B)$ is unknown, then a short-cut to recruiting fragments to B is to compute $\mathcal{F}_A(B) := \{F \in \mathcal{F}(A) \mid F \text{ hits } B\}$ instead of $\mathcal{F}(B)$, at the price of obtaining a less comprehensive analysis. Alternatively, one could first compare the

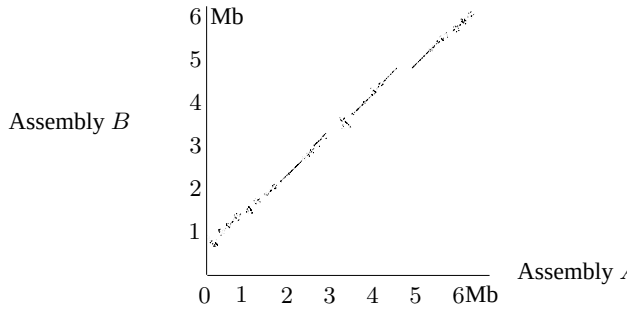

Fig. 2. Fragment based dot-plot comparison of two different assemblies of a 6Mb region of chromosome 2 in human. Each point represents a fragment that hits both assemblies.

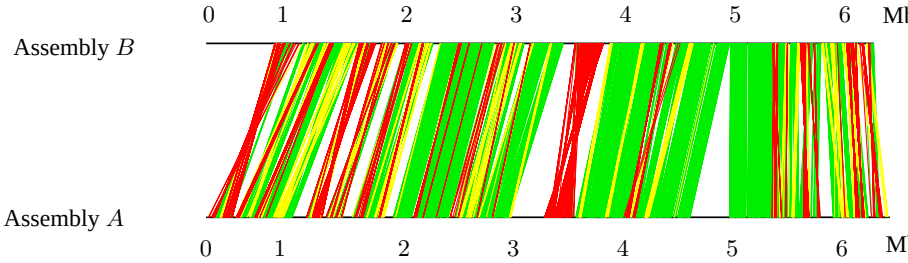

Fig. 3. Fragment Based Line-Plot Comparison. Each line segment represents a fragment that hits both assemblies. Medium grey lines represent fragments contained in the heaviest common subsequence (HCS) of consistently ordered and oriented segments, light grey lines represent consistently oriented segments that are not contained in the HCS, and dark grey lines represent fragments (or segments) that have opposite orientation in the two assemblies.

consensus sequence of assembly B directly against that of assembly A and then project fragments from A onto B wherever compatible with the segments of local alignment between A and B.

2.3 Fragment Segmentation

For analysis purposes and also to speed up visualization significantly, it is useful to segment the fragment matches by determining the maximal consistent and consecutive runs of them.

Consider a fragment $F \in \mathcal{F}(A, B)$. We say that F has *preserved orientation*, if and only if F has the same orientation in A and B, i.e., if either both $s(F, A) < t(F, A)$ and $s(F, B) < t(F, B)$, or both $s(F, A) > t(F, A)$ and $s(F, B) > t(F, B)$ hold. Let $\mathcal{F}^+(A, B)$ denote the set of all fragments that have preserved orientation and set $\mathcal{F}^-(A, B) := \mathcal{F}(A, B) \setminus \mathcal{F}^+(A, B)$.

For any two fragments $F, G \in \mathcal{F}(A, B)$, define $F <_A G$, if $s(F, A) < s(G, A)$, and define $F <_B G$, if $s(F, B) < s(G, B)$. Because we assume that all s values are distinct, these are both total orderings and we use $\text{pred}_A(F)$ and $\text{succ}_A(F)$ to denote the $<_A$-predecessor and $<_A$-successor of F, respectively.

A sequence $S = (F_1, F_2, \ldots, F_k)$ of fragments is called a *matched segment*, in either of the two following cases:

1. $\{F_1, F_2, \ldots, F_k\} \subseteq \mathcal{F}^+(A, B)$ and $\text{succ}_A(F_i) = \text{succ}_B(F_i)$ for all $i = 1, 2, \ldots, k-1$, or
2. $\{F_1, F_2, \ldots, F_k\} \subseteq \mathcal{F}^-(A, B)$ and $\text{succ}_A(F_i) = \text{pred}_B(F_i)$ for all $i = 1, 2, \ldots, k-1$.

A matched segment is called *maximal*, if it can't be extended.

Let $\mathcal{S} := \mathcal{S}(\mathcal{F}(A, B)) = \{S_1, S_2, \ldots, S_n\}$ denote the set of all maximal matched segments of $\mathcal{F}(A, B)$, and let $\mathcal{S}^+$ and $\mathcal{S}^-$ denote the subset of such segments in cases 1 and 2, respectively. Both $\mathcal{S}^+$ and $\mathcal{S}^-$ can be computed in a simple loop that considers each fragment in $<_A$ order and decides whether it extends the current segment or defines the start of a new one.

The *A-support* of a matched segment $S = (F_1, F_2, \ldots, F_k)$ is defined as the interval $[s(S, A), t(S, A)]$, with $s(S, A) := \min_{F \in S}(s(F, A), t(F, A))$ and $t(S, A) := \max_{F \in S}(s(F, A), t(F, A))$. The *B-support* is defined similarly. Let $\text{len}(S)$ denote the minimum length of the A- and B-supports of S.

2.4 Heaviest Common Subsequence

Given two orderings O_1 and O_2 of the set of numbers $\{1, 2, \ldots, n\}$ (for some fixed number n) and a weight function $w : \{1, 2, \ldots, n\} \to \mathbb{N}^{\geq 0}$. A subsequence $H := H(O_1, O_2, w)$ of both orderings is called a *heaviest common subsequence*, if it has maximal weight $w(H) := \sum_{h \in H} w(h)$. The heaviest common subsequence can be computed in $O(n \log n)$ time and space, see [3].

For $\mathcal{S} = (S_1, S_2, \ldots, S_n)$, let O_1 and O_2 denote the ordering of the indices $1, 2, \ldots, n$ induced by the orderings of $\mathcal{S}$ defined by $s(\cdot, A)$ and $s(\cdot, B)$, respectively. With weight function $w(i) := \text{len}(S_i)$, compute the heaviest common subsequence H of O_1 and O_2.

We call $\mathcal{H} := \{S_i \in \mathcal{S} \mid i \in H\}$ the *heaviest common subsequence of matched segments*. We can distinguish between four categories of matched segments:

1. $\mathcal{S}^+ \cap \mathcal{H}$ is the set of segments that have the same ordering and orientation in both assemblies,
2. $\mathcal{S}^- \cap \mathcal{H}$ is the set of segments that have the same position in both assemblies, but are inverted with respect to each other,
3. $\mathcal{S}^+ \setminus \mathcal{H}$ is the set of segments that have transposed positions, and
4. $\mathcal{S}^- \setminus \mathcal{H}$ is the set of segments that appear both transposed and inverted.

The amount of sequence contained in each of these four categories is a good measure of how similar two assemblies are. In visualization, using different colors for each of them significantly enhances the dot-plot and line-plot representation described above, see Figure 3.

2.5 Segment Displacement

Consider two segments $S = (F_1, F_2, \ldots)$ and $T = (G_1, G_2, \ldots)$. We say that S and T are *parallel* if either both $s(F_1, A) < s(G_1, A)$ and $s(F_1, B) < s(G_1, B)$, or both $s(F_1, A) > s(G_1, A)$ and $s(F_1, B) > s(G_1, B)$ hold.

It seems reasonable to "trust" those portions of the two assemblies that are covered by segments from the heaviest common subsequence $\mathcal{H}$. Thus, we propose to measure the amount by which the positioning of a segment S not in $\mathcal{S}^+ \cap \mathcal{H}$ differs in the two assemblies as follows: We define the *displacement* $D(S)$ associated with S as the sum of lengths of all segments in $\mathcal{H}$ that are not parallel to S. In Figure 4 we plot segment length vs. segment displacement.

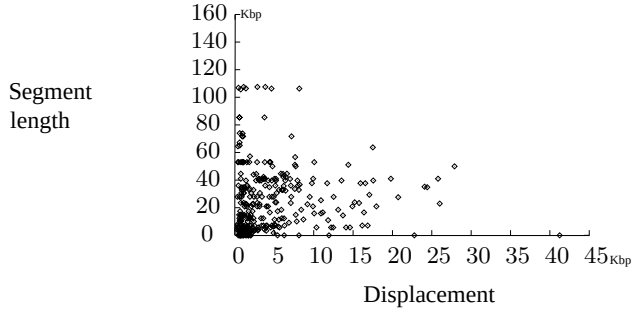

Fig. 4. Scatter-Plot Comparison of Two Assemblies: a dot (x, y) represents a sequence segment S of length $\text{len}(S) = x$ whose displacement $D(S)$ is y. In other words, the placement of S in the two assemblies differs by at least $D(S)$ bp. Note that points along the x-axis correspond to in-place inversions.

3 Mate-Pair-Based Evaluation Methods

Let A and B be two assemblies of a chromosome and let $\mathcal{F}$ be a set of associated fragments. Assume now that the fragments in $\mathcal{F}$ were generated using a "double-barreled" shotgun protocol in which *mate-pairs* of fragments are obtained by reading both ends of longer clones. For purposes of this paper, a *mate-pair library* $M = (L, \mu, \sigma)$ consists of a list L of pairs of *mated* fragments, together with a mean estimate μ and standard deviation σ for the length of the clones from which the mate-pairs were obtained, see Figure 5.

Typical clone sizes used to produce mate-pair libraries used in Celera's human genome sequencing were 2kb, 10kb, 50kb, and 150kb. The quality of shorter mate-pairs can be very good with a standard deviation of about 10% of the mean length, whereas the standard deviation can reach 20% for long clones. Also, because both ends of clones are read in separate sequencing reactions, there is a potential for mis-associating mates.

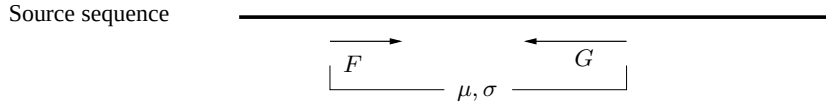

Fig. 5. Two fragments F and G that form a mate-pair with known mean distance μ and standard deviation σ. Note their relative orientation in the source sequence.

However, a high level of automation and electronic sample tracking can reduce the occurrences of this problem to below 1%. By construction, any fragment will occur in at most one mate-pair.

Given an assembly A with fragments $\mathcal{F}(A)$ and a collection of mate-pair libraries $\mathcal{M} = \{M_1, M_2, \ldots\}$, let $m = \{F, G\} \subset \mathcal{F}(A)$ be a mate-pair occurring in some library $M_i = (L, \mu, \sigma)$. Then m is called *happy* if the positioning of F and G in A is reasonable, i.e., if F and G are oriented towards each other (as in Figure 5) and $| \, |s(F, A) - s(G, A)| - \mu| \leq 3\sigma$, say. An unhappy mate-pair m is called *mis-oriented* if the former condition is not satisfied, and *mis-separated* if only the latter condition fails.

3.1 Clone-Middle Plot

We obtain a *clone-middle plot* for A as follows: For each pair of fragments $F, G \in \mathcal{F}(A)$ that occurs in a mate-pair library M, draw a line segment from $(t(F, A), y)$ to $(t(G, A), y)$, where $y \in [0, 1]$ is a randomly chosen height. Lines can be shown in different colors depending on whether the corresponding mate-pair is happy, mis-separated or mis-oriented, see Figure 6, and also Figure 6 in [7]. The interval $[t(F, A), t(G, A)]$ (assuming w.l.o.g. $t(F, A) < t(G, A)$) is called the *clone-middle* (in A) associated with the pair F, G.

One draw-back of this visualization for large assemblies is that substantially misplaced pairs give rise to very long lines in the plot and obscure the view of local regions. To address this, we introduce the *localized* clone-middle plot (see Figure 7): Let $\{F, G\}$ be a mis-separated or mis-oriented mate from some library $M = (L, \mu, \sigma)$. Assume w.l.o.g. that $s(F, A) < s(G, A)$. Represent the mate-pair by a line that indicates the range in which F expects to see G, i.e., by drawing a line segment from $t(F, A)$ of length $\mu + 3\sigma - (\text{len}(F) + \text{len}(G))$ towards the right, if $s(F, A) < t(F, A)$, and to the left, otherwise. As above, define the *clone-middle* accordingly.

Mis-separated and mis-oriented mate-pairs indicate discrepancies between a given assembly and the original source sequence or chromosome, as follows.

3.2 Breakpoint Detection

Loosely speaking, a *breakpoint* of an assembly A is a position p in A such that the sequence immediately to the left and right of p in A comes from two separate regions of the source sequence.

Let $m = \{F, G\}$ be a mis-oriented mate-pair such that $s(F, A) < s(G, A)$. We distinguish between three different cases: *normal*-oriented: both fragments are oriented to the right; *anti*-oriented: both are oriented to the left; and *outtie*-oriented: F is oriented

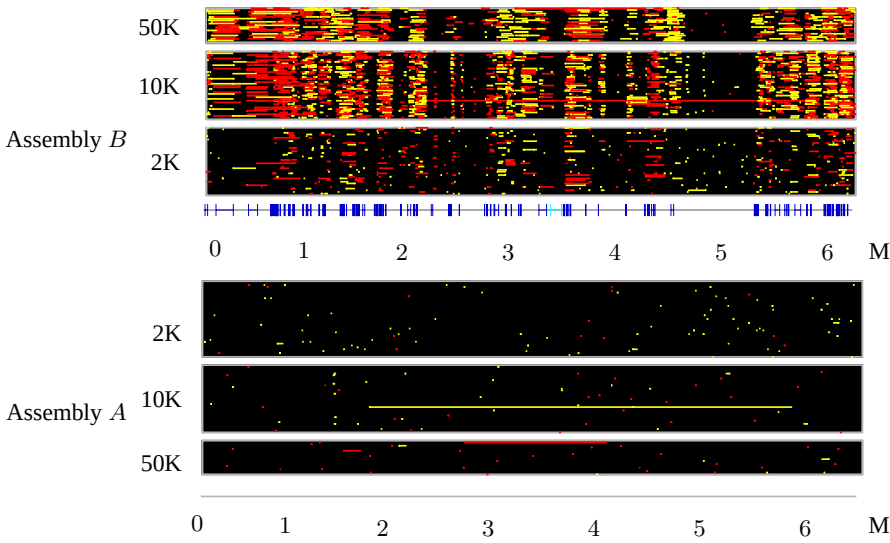

Fig. 6. Clone-Middle Diagram for Assemblies A and B. Each mate-pair m is represented by a horizontal line segment joining its two fragments, if m is mis-separated (shown in light grey) or mis-oriented (shown in dark grey). Happy mates are not shown. Mate-pairs are grouped by "library", labeled $2K$, $10K$ and $50K$. Ticks along the axis indicate putative breakpoints, as inferred from the mis-oriented mates.

to the left and G is oriented to the right. (Happy and mis-separated mates are *innie*-oriented).

We now describe a simple but effective heuristic for detecting breakpoints. Choose a threshold $T > 0$, depending on details of the sequencing project. (All figures in this paper were produced using $T = 5$.) An *event* is a three-tuple (x, t, a) consisting of a coordinate $x \in \{1, \ldots, \text{len}(A)\}$, a type $t \in \{\text{normal}, \text{anti}, \text{outtie}, \text{mis-separated}\}$, and an "action" $a \in \{+1, -1\}$, where $+1$ or -1 indicates the beginning or end of a clone-middle, respectively. We maintain the number of currently *alive* mates $V(t)$ of type t. For each event $e = (x, t, a)$ in ascending order of coordinate x: If $a = +1$, then increment $V(t)$ by 1. In the other case ($a = -1$), if $V(t) \geq T$, then report a breakpoint at position x and set $V(t) = 0$, else decrease $V(t)$ by 1. (For a better estimation of the true position of the breakpoint, report the interval $[x', x]$, where x' is the coordinate of the most recent alive $+1$-event of type t.) Breakpoints estimated in this way are shown in Figure 7.

A useful variant of the breakpoint estimator is obtained by taking the current number of alive happy mates into account: Scanning from left to right, a breakpoint is said to be present at position x if there exists an event $e = (x, t, -1)$ such that the number of alive unhappy mates of type t exceeds the number of alive happy mates of type t.

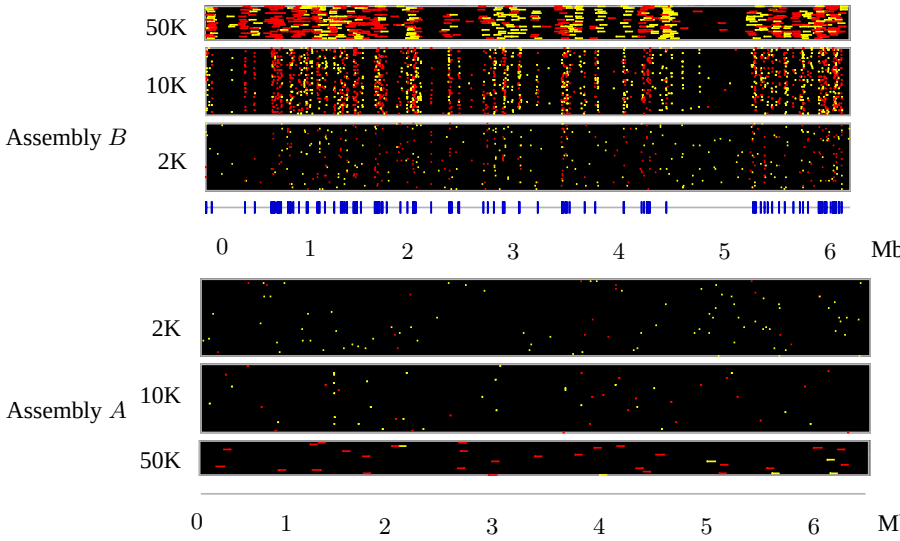

Fig. 7. A Localized Clone-Middle Diagram for Assemblies A and B. Here, each mis-separated or mis-oriented mate-pair is represented by a line that indicates the expected range of placement of the right mate with respect to the left one. Ticks along the axis indicate putative breakpoints, as inferred from the mis-oriented mates.

3.3 Clone-Coverage Plot

Similar to the fragment-coverage plot discussed in Section 2, one can use the clone-coverage events to compute a *clone-coverage* plot for each of the types of mate-pairs, see Figure 8.

Note that the simultaneous occurrence of both high happy and high mis-separated coverage may indicate the presence of a polymorphism in the fragment data.

3.4 Synthesis

Combining all the described methods into one view gives rise to a tool that is very helpful deciding by how much two different assemblies differ and, more, which one is more compatible with the given fragment and mate-pair data; see Figure 9. This latter capability is an especially powerful aspect of analysis in terms of fragments and mate-pairs.

4 Some Applications

The techniques described in this paper have a number of different applications in comparative genomics. Originally, our goal was to design a tool for comparing the similarities and differences of assemblies of human chromosomes produced at Celera with

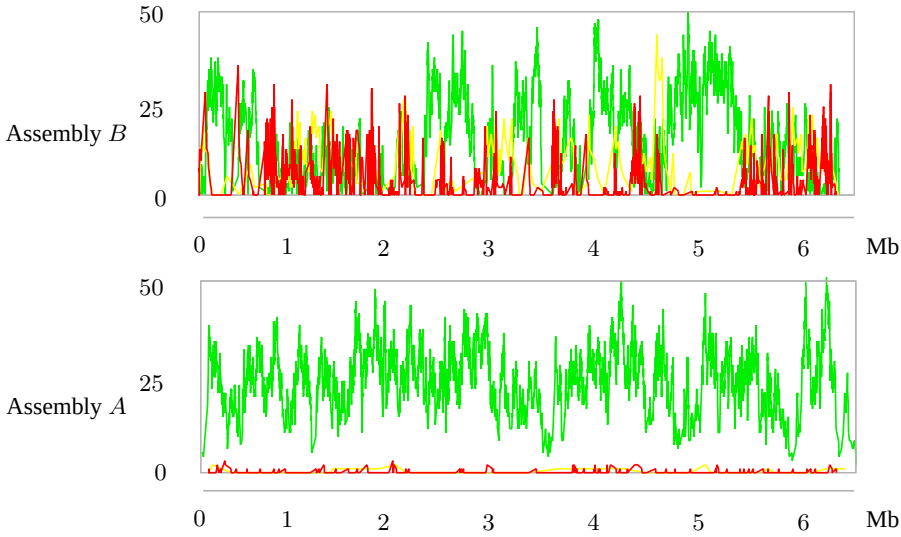

Fig. 8. Clone-coverage plot for assemblies A and B, showing the number of of happy mate-pairs (medium grey), mis-separated pairs (light grey) and mis-oriented ones (dark grey).

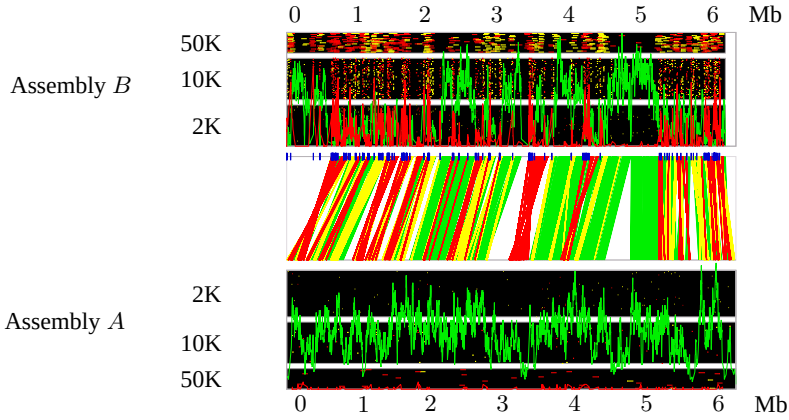

Fig. 9. A combined line-plot, clone-middle, clone-coverage and breakpoint view of the two assemblies A and B indicates that assembly A is significantly more compatible with the given fragment and mate-pair data than assembly B is.

those produced by the publicly funded Human Genome Project (PFP). A detailed comparison based on our methods is shown in Figures 6 and 7 of [7]. As an example, we show the comparison for chromosome 2 in Figure 10. For clarity, only segments of length 50kb or more are shown.

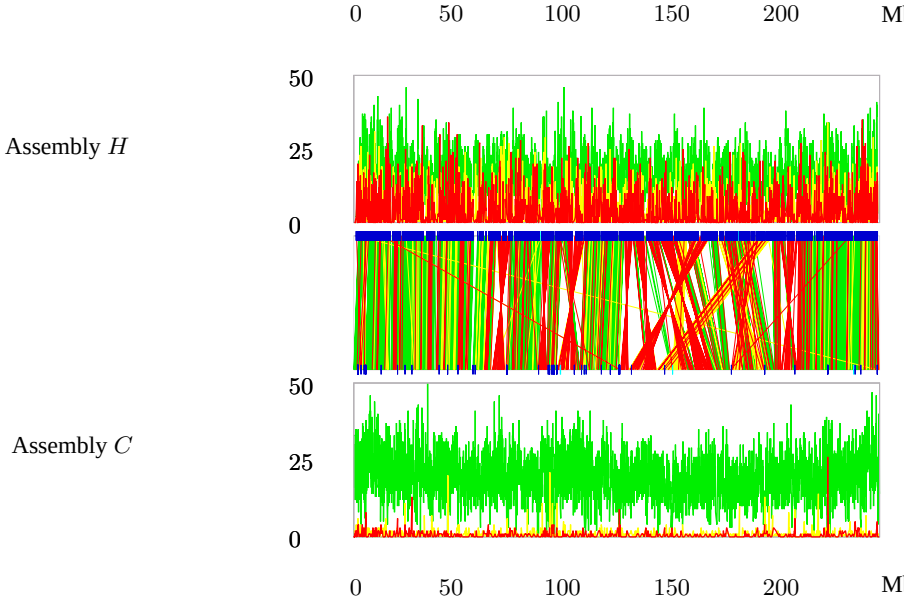

Fig. 10. Line-plot and breakpoint comparison of two different assemblies of chromosome 2 of human. Assembly C was produced at Celera [7] and assembly H was produced in the context of the publicly funded Human Genome Project and was released on September 5, 2000 [2]. The number of detected breakpoints (indicated as ticks along the chromosome axes) is 73 for C and 3592 for H.

4.1 Feature-Tracking

A second application is in tracking forward features from one version of an assembly to the next. To illustrate this, we consider two assemblies of chromosome 19 produced in the context of the PFP from publicly available data. Assembly H_1 was released on September 5, 2000 and assembly H_2 was released on January 9, 2001 [2].

How much did the assembly change and did it improve? The line-plot comparison of H_1 and H_2 in Figure 11 indicates that many local changes have taken place. A detailed analysis (not reported here) shows that many changes are due to a change of orientation of so-called "supercontigs" in the assembly. The number of detected breakpoints dropped from 723 to 488.

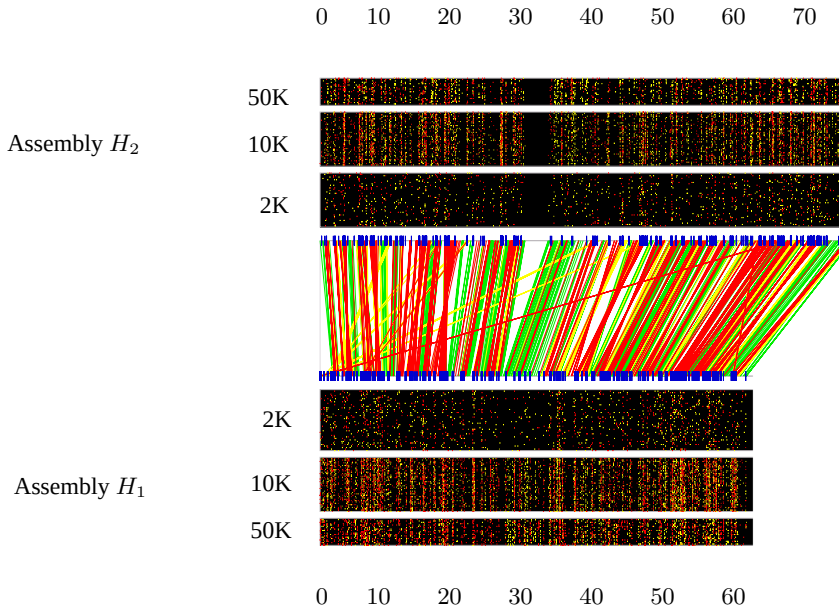

Fig. 11. Line-plot, clone-middle and breakpoint comparison of the PFP assembly H_1 of chromosome 19 as of September 5, 2000, and the a more recent PFP assembly H_2 dating January 9, 2001.

4.2 Comparison of Different Chromosomes

Additionally, our algorithms can be used to compare different chromosomes of the same species e.g. in search of duplication events, but also to compare different chromosomes from different species, in the latter case using a lower stringency alignment method to define fragment hits.

We illustrate this by a comparison of chromosome X and Y of human, as described in [7]. In this analysis we use only uniquely hitting fragments. In summary, we see approximately 1.3Mb of sequence in conserved segments, of which 164kb are contained in the heaviest common subsequence (relative to the standard orientation of X and Y), 82kb are contained in other segments of the same orientation and 1.05Mb in oppositely oriented segments, see Figure 12. We observe orientation preserving similarity at both ends of the chromosomes and a large inverted conserved segment in the interior of X.

References

1. A. Edwards and C.T. Caskey. Closure strategies for random DNA sequencing. *METHODS: A Companion to Methods in Enzymology*, 3(1):41–47, 1991.
2. D. Haussler. Human Genome Project Working Draft. `http://genome.cse.ucsc.edu`.

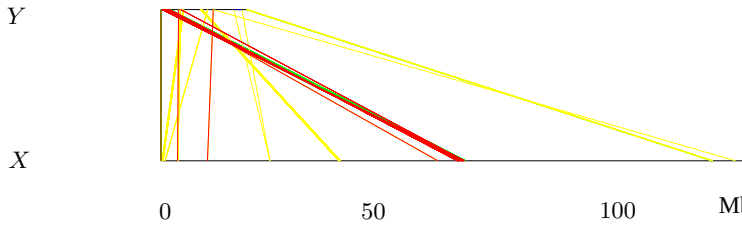

Fig. 12. A line-plot comparison of chromosome X vs. Y of human showing segments of highly conserved non-repetetive sequence.

3. G. Jacobson and K.-P. Vo. Heaviest increasing/common subsequence problems. In *Proceedings 3rd Annual Symposium on Combinatorial pattern matching (CPM)*, pages 52–66, 1992.
4. E. W. Myers, G. G. Sutton, A. L. Delcher, I. M. Dew, D. P. Fasulo, M. J. Flanigan, S. A. Kravitz, C. M. Mobarry, K. H. J. Reinert, K. A. Remington, E. L. Anson, R. A. Bolanos, H-H. Chou, C. M. Jordan, A. L. Halpern, S. Lonardi, E. M. Beasley, R. C. Brandon, L. Chen, P. J. Dunn, Z. Lai, Y. Liang, D. R. Nusskern, M. Zhan, Q. Zhang, X. Zheng, G. M. Rubin, M. D. Adams, and J. C. Venter. A whole-genome assembly of Drosophila. *Science*, 287:2196–2204, 2000.
5. F. Sanger, A. R. Coulson, G. F. Hong, D. F. Hill, and G. B. Petersen. Nucleotide sequence of bacteriophage λ DNA. *J. Mol. Bio.*, 162(4):729–73, 1992.
6. F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.
7. J. C. Venter, M. D. Adams, E. W. Myers, et al. The sequence of the human genome. *Science*, 291:1145–1434, 2001.