IS WHOLE HUMAN GENOME SEQUENCING FEASIBLE?*

Eugene W. Myers(1) and James L. Weber(2)

(1) Department of Computer Science
University of Arizona
Tucson, AZ 85721-0077

(2) Center for Medical Genetics
Marshfield Medical Research Foundation
Marshfield, WI 54449

1. SHOTGUN SEQUENCING AND ITS VARIATIONS

1.1 Basic Shotgun Sequencing:

As the time to sequence the entire human genome approaches, it appears that the predominant method of collecting this data will entail some variation of the shotgun sequencing strategy [1]. The startpoint for the basic shotgun experiment is a pure sample of a large number of copies of a particular stretch of DNA of some length *G*, say 100,000 base pairs (100Kbp) for the purposes of illustration. The sample is either sonicated or nebulated, randomly partitioning each copy into pieces called *inserts*. The resulting pool of inserts is then size-selected so that fragments that are too large or too small are removed from further consideration. The inserts that remain thus represent a random sampling of segments of the source sequence of a given approximate size. A sub-sample of the inserts are then cloned via insertion into a viral phage, called a *vector*, and subsequent infection of a bacterial host. The cloning process results in the production of a pure sample of a given insert so that it may then be sequenced. Typically this is done via the method of Sanger et al. [2] which produces a ladder-like pattern on an electropheretic gel. Generally, only the first 300 to 800 base pairs of the insert can be interpreted from this experiment. This data is called a *read* and is a contiguous subsequence of the source sequence.

*This work was supported in part by NLM Grant R01 LM04960, and DOE Grant DE-FG03-94ER61911.

(1) Corresponding author: gene@cs.arizona.edu, +1 (520) 621-6612, FAX: +1 (520) 621-4246.

For a source sequence of length $G = 100 Kbp$, an investigator might collect, say $R = 1250$ reads, where the average length of the reads is typically $\overline{L}_R = 400$. In summary, one make think of the shotgun approach as delivering a collection of R reads that constitute a random sample of contiguous subsequences of the source sequence of length approximately $\overline{L}_R$.

1.2 The Fragment Assembly Problem:

Given the reads obtained from a shotgun protocol, the computational problem, called *fragment assembly*, is to infer the source sequence given the collection of reads. For our running example, note that altogether $N = R\overline{L}_R = 500 Kbp$ basepairs of data have been collected. Thus one has on average sequenced every basepair in the source $\overline{c} = N/G = 5$ times. The quantity $\overline{c}$ is called the average sequencing redundancy. In practice, an investigator will decide on a given level of redundancy and then clone and sequence inserts until a total of $N = G\,\overline{c}$ base pairs of data have been collected. In designing algorithms and software for fragment assembly one must account for the following characteristics of the data:

- *Incomplete Coverage*: Not every source basepair is sequenced exactly $\overline{c}$ times due to the stochastic nature of the sampling, thus some portions of the source may be covered by more than $\overline{c}$ reads and others may not be covered at all. In general, there can be several such *gaps* or maximal contiguous regions where the source sequence has not been sampled. Gaps necessarily dictate a fragmented, incomplete solution to the problem.

- *Sequencing Errors*: The gel-electropheretic experiment yielding a read, like most physical experiments, is prone to error especially near the end of a read where the blots of the ladder-like gel pattern are compressed together. Typically the *maximum* error rate ε is in the range of 5-10%.

- *Unknown Orientation*: The source sample is double-stranded DNA. Which of the two strands is actually read depends on the arbitrary way the given insert oriented itself in the vector. Thus one does not know whether to use a read or its Watson-Crick complement[†] in the reconstruction.

In most common systems for fragment assembly the problem is solved in three phases [3,4]. In the first *overlap phase*, every read is compared against every other read and its complement to determine if they overlap. Due to the presence of sequencing errors, an overlap is necessarily approximate in that not all characters in the overlapping region coincide. This problem is a variation on traditional sequence comparison where the degree of difference permitted is bounded by ε. In the second *layout phase*, the relative position of each read is determined by selecting an approximate overlap that links it to the rest. That is, a spanning forest of the graph of all overlaps is selected, in effect producing an arrangement or layout of the fragments. The selection of the spanning forest is driven by some optimization criterion for the layout, most often to minimize length [4], but maximizing likelihood has also been advocated [5]. Given a layout of the fragments, the final *consensus phase,* produces a multi-alignment of the reads in regions where

[†] The Watson-Crick complement $(a_1 a_2 \cdots a_n)^c$ of a sequence $a_1 a_2 \cdots a_n$ is $wc(a_n) \cdots wc(a_2) wc(a_1)$ where $wc(A) = T$, $wc(T) = A$, $wc(C) = G$, and $wc(G) = C$.

the coverage is 3 or greater. The multi-alignment results in a *consensus sequence* which is reported as the reconstructed source sequence.

1.3 Current Computational Developments:

Ultra-large Problems. As sequencing efforts have scaled up for human genome sequencing, some new facets of the problem have become evident. The first fact is the rapid increase in the size G of the source sequences that are being shotgunned. In the early 1980's, when shotgun sequencing was first proposed, a typical source sequence size was $3Kbp$. By 1990, experimentalists were shotgun sequencing an entire cosmid at a time for which $G \approx 40Kbp$. In 1995, Ventner reported shotgun sequencing the entirety of *H. Influenza* for which $G \approx 1,800Kbp$. It is not clear how far this trend will continue, but the size of the source sequences being directly shotgunned has far exceeded any reasonable projection that would have been made five years ago. It is thus clear that one must design algorithms that work on ultra-large problems: ones so large that memory and time requirements become significant concerns.

Repeats. Another significant development is the extent to which repetitive elements occur within a target sequence. Such occurrences were relatively rare previously because (a) G was small, and (b) the inherently simpler genomes of low-order organisms were the source. It has now become apparent that high-order genomes, such as that of humans, have a rich repeat structure. Combined with the very large stretches being tackled in a single shotgun effort, the chance of encountering not one but several repetitive substructures in the source is almost certain. Moreover repeats occur at several scales. For example, in the T-cell receptor locus of humans there is a 5-fold repeat of a trypsinogen gene that is $4Kbp$ long and varies 5-10% between copies. Three of these were close enough together that they appeared in a single shotgun-sequenced cosmid source [6]. Such large scale repeats are problematic for shotgun approaches as reads with unique portions outside the repeat cannot span it. Smaller elements such as palindromic Alu's of length approximately $300bp$, do not share this feature but are still problematic as they can constitute up to 50-60% of the target sequence [7,8]. Finally, in telomeric and centromeric regions, micro-satellite repeats of the form x^n, where the repeated string x is three to six bases long and n is very large, are common [7]. Thus one must design software solutions that can resolve assemblies involving repetitive DNA.

Constraints. While experimentalists have ambitiously increased the size of the source sequences, the technology for obtaining a read has not improved the average length of a read $\bar{L}_R$ at a corresponding rate, leading to smaller and smaller ratios of $\omega = \bar{L}_R / G$. A seminal paper by Waterman and Lander [9] showed that the expected number of gaps (unsampled regions) grows as $Re^{-\omega R}$. Thus the fragmentation of the solution into a collection of gap-separated "contigs", giving the covered intervals of the source, is increasing significantly. This combined with the increasing difficulty of correctly resolving repetitive elements in the source, has induced investigators to develop hybrid or alternative strategies to reduce these problems. Some examples are as follows:

- *PCR Gap Closure*: While gaps can be frequent it is also the case that they are generally quite small, $150bp$ or less. Thus a common technique is to shotgun sequence up to a given redundancy and then "close" gaps by designing PCR probes to amplify the sequence in the gap [10]. The reads obtained from the PCR experiments are known to overlap with reads adjacent to the gaps.

- *Double-Barreled Shotgun Sequencing*: Inserts are size selected so that the average length $\overline{L}_I$ is roughly $2\overline{L}_R$ or longer and *both* ends of the insert are sequenced [11]. This gives rise to a pair of reads that are known (1) to be in opposite orientations and (2) to be at a distance from each other approximately equal to the insert length.
- *Transposon-Mapped Sequencing*: A carefully engineered transposon is repeatedly inserted into the source and a map of the insertion location is created via restriction enzyme digestion of the unique cut site built into the transposon. Then ladder sequencing is performed in both directions from an inserted transposon with the aid of primers built into both ends of the transposon [12]. This gives rise to a set of overlapping reads as shown in Figure 1. In this scenario it is known that the pair of reads emanating from a given transposon (1) overlap by 4 or 5 base pairs (due to a replication that occurs during transposon insertion), (2) are in the opposite orientation, and (3) that the forward read from transposon i overlaps that of the reverse read from transposon $i+1$.

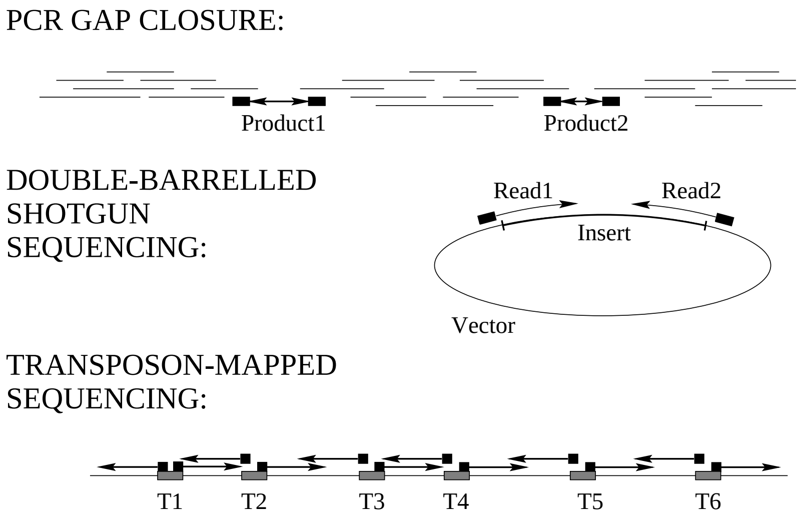

Figure 1: Sequencing protocols involving constraints on the relative positions of reads in terms of overlap, orientation, and distance.

In all three cases, there is additional information about how the reads should be put together in a layout. We have proposed that this information should be specified to an assembler as a collection of orientation, overlap, and/or distance constraints that we have demonstrated are sufficient to describe all protocols in practice today [13,14]. Regardless of the method of specification, it is clear that modern assembly software will have to handle shotgun data that is augmented by additional constraint information.

1.4 Variations on Shotgun Sequencing:

In the previous subsection, we began to explore variations on the basic shotgun strategy, the objectives of which were to alleviate the problems of achieving closure and of resolving sequence repeats for large scale projects on higher-order organisms. Before exploring these further some definitions, illustrated in Figure 2, are required. A *contig* is a collection of reads that cover a contiguous region of the source. Note that contigs are mutually non-overlapping being separated from each other by *gaps*. Furthermore, without additional information it is impossible to tell how to order the contigs along the source sequence. Consider next, projects in which at least some percentage of the reads are from opposing ends of suitably long inserts. Often these pairs are in different contigs, which consequently, orders them and gives the approximate distance between them. A maximal collection of contigs so linked is termed a *scaffold* [15]. Note, as illustrated in Figure 2, that scaffolds can and do overlap with the contigs of one scaffold interspersed between those of another. Thus one cannot sensicaly speak of "gaps" between scaffolds. Instead the relevant property of scaffolds is the frequency with which regions of the genome are linked by one.

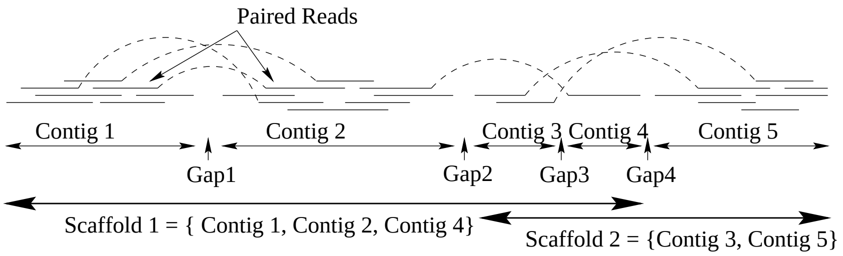
Paired Reads
Contig 1
Contig 2
Contig 3
Contig 4
Contig 5
Gap1
Gap2
Gap3
Gap4
Scaffold 1 = { Contig 1, Contig 2, Contig 4}
Scaffold 2 = {Contig 3, Contig 5}

Figure 2: Illustration of contigs, gaps, and scaffolds.

Double-Barreled Shotgun Sequencing: In a "pure" version of this approach each of I inserts of average length $\overline{L}_I$ is end-sequenced. One can then define the *map redundancy* [15] of the project as $\bar{m} = I\overline{L}_I/G$. Now in general $\bar{m}$ is larger than the sequence redundancy $\bar{c} = \bar{m}(2\overline{L}_R/\overline{L}_I)$ when $\overline{L}_I \geq 2\overline{L}_R$, i.e., when the inserts are size selected so the reads don't overlap. Thus the expected number of gaps between *insert* contigs, $\frac{1}{2}Re^{-\sigma\frac{1}{2}R}$ where $\sigma = \overline{L}_I/G$, is progressively smaller the larger the insert size $\overline{L}_I$. In other words, double-barreled shotgun sequencing tends to produce very large scaffolds. This positions sequence contigs with respect to each other, and so facilitates the determination of PCR primer pairs needed for gap closure. Furthermore, for sufficiently long inserts, it is unlikely that both reads of an insert will lie in a large scale repeat of the source. Thus the read in the relatively unique portion of the source effectively determines which copy of a repeat its mate is in. Because of this, double-barreled shotgun sequencing has been found to be a very powerful way of resolving repeats.

Recent simulation studies [15] have indicated that from a purely informatic perspective there is advantage in using longer inserts and no advantage in having some percentage of the reads be unpaired. However, this must be tempered against the experimental fact, that because of the different cloning vehicles involved as the insert becomes larger (e.g. plasmid → λ → cosmid → PAC,BAC,P1 → YAC), it is more difficult to sequence the ends of long inserts and there is greater chance of chimerism (two or more inserts ending up in the same cloning vehicle). Thus there is counter-balancing economic pressure to use single reads and shorter inserts. Fortunately one loses little of the benefits of having long end-sequenced inserts in hybrid schemas where a sizable fraction of a project is single reads, and where the paired reads are from inserts over a distribution of insert lengths skewed to the shorter lengths.

Ordered Shotgun Sequencing (OSS): While one may have several repeats in a long source sequence and a shotgun assembly may have many gaps, it is rare that either of these problems occur for source sequences on the order of the length of a λ-clone, e.g. $G \approx 10Kbp$. To take advantage of this observation, Chen et. al [16] devised the following two-tiered approach to sequence a large source sequence of length $G \gg 10Kbp$. For $\bar{m} \approx 10$, begin by sequencing both ends of $I = G\bar{m}/\bar{L}_I$ λ-sized inserts whose average length is $\bar{L}_I \approx 10Kbp$. Initially, let the current fragment set be the set of reads sequenced above, and set the sequenced intervals of the source to the empty set. Repeat the series of steps below until nothing new can be added to the sequenced intervals.

(a) Assemble the current set of fragments and sequenced intervals, treating each sequenced interval as a single read.

(b) Select from the assembly of (a) the scaffold that spans the largest unsequenced interval of the source. For the first iteration, this criterion reduces to selecting the longest scaffold.

(c) Select the smallest subset of inserts of the scaffold of (b) that cover the unsequenced interval, i.e., select a *minimum tiling set*.

(d) Shotgun sequence each insert in the minimum tiling set of (c).

(e) Use the sequence determined in (d) to update the sequenced intervals. Remove all reads involved in the scaffold of (b) from the current set of reads as well as all those now discovered to be completely within a sequenced interval of the source.

Figure 3 illustrates this process. Note that with each round of the steps above, new overlaps with end reads occur because the interior of several λ-inserts is sequenced. Because the map redundancy is high, it is thus very likely that the process will result in the sequence of all of the source. As a final point, not that while towards the later rounds, it is typical to discover maximal scaffolds extending from the ends of distinct sequenced intervals. One may pursue all such scaffolds in a single round as their tiling sets cannot overlap (e.g. see "Round 3" of Figure 3).

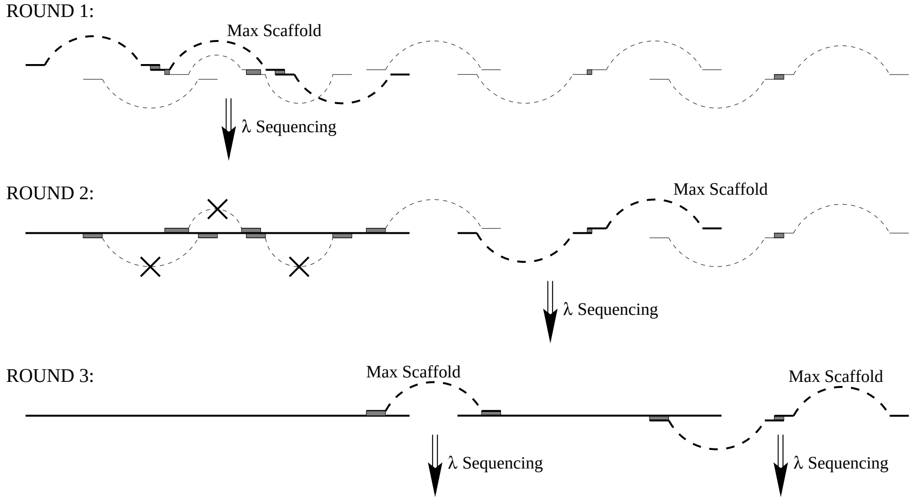

Figure 3: Illustration of Ordered Shotgun Sequencing.

Whole Genome Ordered Shotgun Sequencing: Recently some investigators have effectively proposed using the OSS strategy on the scale of the entire human genome. In this case the size of the source is $G = 3{,}000 Mbp$, $\bar{m}$ has been increased to 20, and instead of λ-clones, BACs of size $I = 400 Kbp$ will be end-sequenced. One should see that the problem and potential solutions are identical to the above, save that the proposers are confident that they can shotgun sequence BACs using the double-barreled approach. As proposed investigators would select some seed BACs at random to sequence. Then minimally overlapping BACs at each end of a seed will be found and shotgun sequenced to extend the seed sequence until eventually the entire genome is covered. This is tantamount to OSS where one dispenses with finding maximal scaffolds, and simply proceeds with a pair of extending inserts in each iteration.

Map-Based Whole Genome Sequencing: A final approach to whole genome sequencing is based on the observation that already available today are $100\text{-}200 Kbp$ resolution maps of *sequence tagged sites* (STSs) of the human genome [17]. An STS is a contiguous sequence of the genome of typically $300 bp$ that is unique, sequence-wise, with respect to rest of the genome and which can be consistently be PCR'd with a pair of primers from each end, typically each about $18 bp$ long. The available STS maps order some 15,000 of these sites along each chromosome of the genome along with a rough estimate of the distance between them. Given the linking and repeat resolving power of double-barreled sequencing the following possibility arises. Directly shotgun the entire human genome at a given coverage $\bar{c}$, say 5-10, sequencing both ends of every insert sampled. Applying a fragment assembly algorithm to the resulting data set of roughly 30-60 million reads would likely by a futile exercise. However, given the map of STS markers, the problem of determining the sequence of the genome reduces to the tractable one of finding a scaffold that spans between a pair of markers roughly $100\text{-}200 Kbp$ apart. We call this the *inter-marker assembly problem*. We will show that, for a wide range of insert size mixes, such a scaffold exists over 99% of the time, and when it does, one may then assemble over 99% of the sequence between the markers yielding exactly the same result that one would obtain if one had shotgun sequenced the stretch between the markers at a redundancy of $\bar{c}$. The remainder of this paper will be devoted to supporting the claims just made via simulation and to the presentation of an algorithm for solving the inter-marker assembly problem.

2. A SIMULATION STUDY OF MAP-BASED WHOLE GENOME SEQUENCING

2.1 Nature of the Simulation:

To test the feasibility of the whole genome sequencing approach, we developed a simulator with the following features. For our model of the genome we decided to assume that it contains two sizes of repetitives elements, *short repeats* and *long repeats*, whose lengths and number (expressed as a percent of the genome) are inputs to each simulation trial. The idea is simply that the short repeats model the relatively abundant small interspersed nucleic elements (SINEs) like Alu's and the long repeats model long interspersed nucleic element (LINEs) and the exons of repeated genes. The length of these elements are controlled by the parameters `SINE_LEN` and `LINE_LEN` while the length of the genome for a given trial is specified by the parameter `GENOME_LEN`. The relative abundance of these elements is determined by the parameters `SINE_PERCENT` and `LINE_PERCENT` which specify the percentage of the genome covered by the given element. That is, for a given trial, SINE_PERCENT * GENOME_LEN / SINE_LEN short repeats and LINE_PERCENT * GENOME_LEN / LINE_LEN long repeats are generated and placed with uniform probability across the genome. The generated elements are guaranteed not to overlap but they can and do end up being adjacent to each other. We term the sum, SINE_PERCENT + LINE_PERCENT, of the two percentages, the *repetitiveness* of the genome.

With regard to the sequencing protocol, two types of inserts are assumed to be sequenced: *long inserts* for which both ends are sequenced, and *short inserts* whose length is selected so that end reads overlap by 100 bases, effectively giving a single long read of twice the average read length less 100. The length of a long insert, LONG_LEN, the length of a read, READ_LEN, and the ratio of long to short reads sampled, LS_RATIO, are parameters that can be set for each simulation trial. From these parameters the length of a short insert for a given trial is inferred to be 2*READ_LEN−100. Finally, the sequencing redundancy for a sequencing trial is set by the parameter REDUNDANCY. Given settings for these parameters,

```
.5*REDUNDANCY*GENOME_LEN / ((1+LS_RATIO)*READ_LEN − 50)
```

short inserts are sampled uniformly from the genome, as are

```
.5*REDUNDANCY*GENOME_LEN*LS_RATIO / ((1+LS_RATIO)*READ_LEN − 50)
```

long inserts in a given trial.

In, for example, the analysis of Waterman and Lander it was assumed that every entity was of exactly the length specified by its controlling parameter. However, in reality, insert lengths, read lengths, and repetitive element lengths (of a given type) range over some distribution of lengths. As in Waterman and Lander's paper, we also argue that such variation, if limited to say 15-25% of the average, does not significantly change the outcomes of the simulations. Indeed, some initial experimentation on small models confirmed this. Given the size of the genomes to be simulated, the number of repetitive elements and especially the number of inserts involved is huge and stresses the memory capacity of most machines. Modeling variability in insert length doubles the memory required, and further allowing variability in the read length at each end doubles the memory required yet again. Thus in the production simulator, we adopt the compromise of modeling variability in insert and repetitive element lengths, but assume all reads are of exactly length READ_LEN. Length variation is controlled by the parameter LEN_VARIATION: the length of inserts and repeats is chosen uniformly from the interval 1 ± LEN_VARIATION times the target length.

Given the genome and sequenced inserts for a specific trial, the simulator examines the nature of the contigs, scaffolds, gaps, map connectivity, and other features of the assembly that could result from such data. A *false overlap* between two reads is one where the overlapping portion of their sequences are identical to within error rate ε but the reads were sampled from non-overlapping regions of the underlying genome, i.e., melding the fragments would lead to an incorrect assembly. Note that the perspective in the simulation is distinctly different from that faced by an assembly program operating on real data. For a given trial we know exactly which segment of the genome a read comes from and exactly where every repetitive element is in this genome. Thus the simulator knows exactly which overlaps are true and false, whereas a hypothetical assembler would have to decide this. False overlaps are of two types. In *coincidental overlaps* the overlapping portions of the two reads are from different portions of the genome that are similar just by chance. In *repeat overlaps* the two segments of the genome are similar because they involve copies of the same repetitive element. Whole human genome sequencing would involve at most 75 million fragments, for a total of 10^{16} comparisons. If one insists on at least `MIN_OVERLAP` = 35 basepairs of overlap, then one should expect fewer than one overlap in roughly 10^{15} to 10^{18} to occur coincidentally. That is, apart from computing at most a few coincidental overlaps all other false overlaps computed by a hypothetical assembler would be due to repetitive elements. In subsection 2.3. we will discuss the issue of how an assembler might distinguish repeat overlaps from true overlaps in the context of solving an inter-marker assembly problem. For now we focus on the issue of whether or not there is enough overlaps or connectivity between the reads collected. To that end, we finesse the issue of repeat overlaps by throwing them out along with any true overlaps involving a repeat. That is the simulator will consider read A to *overlap* read B if there are not less than `MIN_OVERLAP` bases of the overlap that are *not within a repetitive element* of the genome. This definition is conservative with respect to connectivity in that the contigs and scaffolds obtained are the sparsest one could imagine from a whole genome data set. This is so as it essentially says that R% of the genome is unusable for connecting fragments in an R% repetitive genome.

An insert is *useless* if it can't possibly be involved in an overlap as defined above. That is, both its reads do not have a segment of `MIN_OVERLAP` or more base pairs lying outside of a repetitive element. Such inserts are removed from further consideration in our simulation analysis. We define the *effective coverage* of the genome as the percentage of the genome covered by the reads of useful inserts. Note that while one read of a long insert may be completely in a repeat, if the other is not, then the insert is useful and both reads are deemed to cover their respective parts of the genome. This is sensible as the exact utility of dual end sequencing is that the end read in the "unique" portion of the genome resolves the repeat that the other end read may find itself in.

Many simulations results are about to be presented. In most plots we show the effect of varying one or two parameters with the remainder being fixed. In such cases it is assumed that the fixed parameters are at the following default values unless otherwise stated:

GENOME_LEN	500Mbp	LONG_LEN	10Kbp
SINE_LEN	300bp	READ_LEN	400bp
LINE_LEN	1500bp	LS_RATIO	1.0
SINE_PERCENT	20%	REDUNDANCY	10.0
LINE_PERCENT	5%	LEN_VARIATION	15%
MIN_OVERLAP	35bp		

Furthermore, in the default case `SINE_PERCENT` will be exactly four times `LINE_PERCENT`, so that when we say a genome is R% repetitive, `LINE_PERCENT = .2*`R and `SINE_PERCENT = .8*`R.

2.2 Simulation Results:

For each simulation trial, we measured effective coverage, $\overline{e}$, the average length or span of a contig, $\overline{ctg}$, the average length of a gap, $\overline{gap}$, the number of contigs formed, C, and the maximum observed gap length, $gap_{\max}$. The first observation, is the confirmation of the following theoretical predictions. Consider holding all parameters constant except for genome length *G*. The observables above behave as follows as a function of *G*:

(1) $\overline{e}(G)$ = a constant = $O(1)$
(2) $\overline{ctg}(G)$ = a constant = $O(1)$
(3) $\overline{gap}(G) = \overline{ctg}(1-\overline{e})\ /\ \overline{e} = O(1)$
(4) $C(G) = G\ /\ (\overline{ctg}+\overline{gap}) = O(G)$
(5) $E[gap_{max]} = O(ln\ G)$

For example, equation (1) says that if at the default parameter settings effective coverage is 99.3% for a genome of say 100Mbp, then it is 99.3% for one of 500Mbp or 3000Mbp. Often this statement is met with surprise as a common misconception of the Lander and Waterman analysis is that one cannot effectively shotgun genomes that are too large. What is actually true is that the number of contigs grows directly proportional to *G* (i.e., equation (4)), thus there is considerable fragmentation in terms of gaps. But the number of gaps that will need to be PCR'd or resolved over a series of 30Kbp cosmids spanning a genome will be the same as the number of gaps that will need to be resolved if one shotgun sequenced the entire genome. The only drawback is that while the average gap length is the same in both scenarios (i.e., equation (3)), the expected value of the maximum observed gap does grow logarithmically (i.e., equation 5 [18]) and so will be an order of magnitude larger in the whole-genome case. Thus one may encounter a few large gaps in the whole genome data set that one does not encounter in a cosmid-by-cosmid or BAC-by-BAC data set. Fortunately, average gap length is generally quite small, e.g., on the order of less than 100bp, when the redundancy is sufficient to guarantee 99% or greater effective coverage. In a whole genome data set we observe a maximum gap of length 2400bp, and a handful over 1000bp. Thus one would expect PCR closure to fail in just a very limited number of cases.

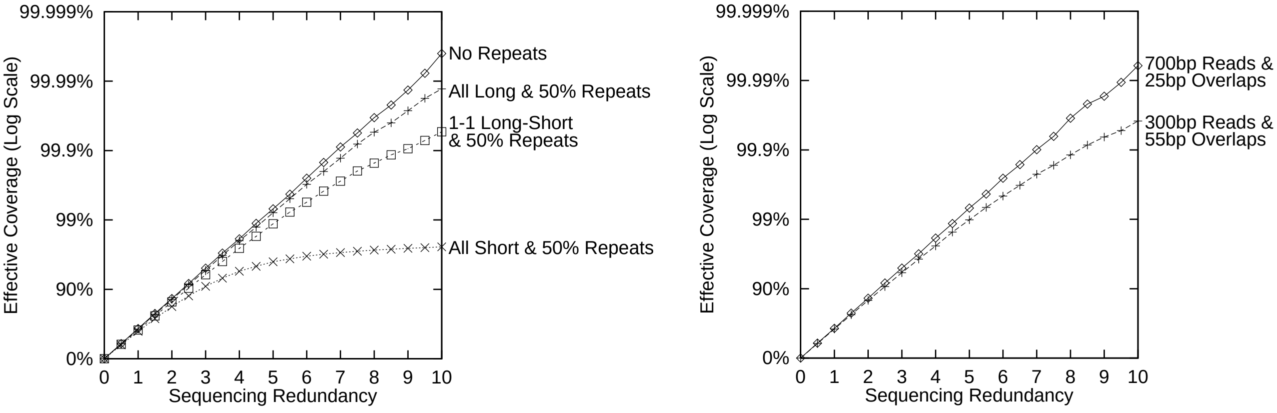

Figure 4: Effective Coverage as a function of (a) repetitiveness and long/short mix, and (b) read length and minimum overlap.

Figure 4a shows a number of plots of expected coverage as function of sequencing redundancy. One plot is for a genome with no repetitive elements, effectively summarizing the

Lander-Waterman analysis. The other three assume a genome that is 50% repetitive implying by our convention for defaults that 40% of the genome is covered by SINEs and 10% by LINEs. For these cases the mix of long-to-short inserts affects the results as short inserts are useless with much greater probability than long inserts. To explain the results, first observe that SINEs of length 300 do not create any obstacle to coverage as they are not long enough to make a read useless (i.e., `READ_LEN ≥ SINE_LEN + 2*MIN_OVERLAP`). However, LINEs of length 1500 cannot be effectively covered by short inserts which at the default settings leaves the inner 10% of each LINE uncovered on average. Thus the ‘‘All short’’ curve tops out at 99% effective coverage. Note, however, that with even a 1-1 ratio of long to short inserts, there is only a slight degradation of coverage compared to the ‘‘All long’’ curve. We may thus conclude that at a sequencing redundancy of 5x or greater and a 1-1 or greater mix of long to short inserts, a whole genome data set will effectively cover more than 99% of the genome unless it contains many more long repetitive elements than are currently being observed in the human genome.

Figure 4b further shows that the assertion above is true for a wide range of read length assumptions and overlap assumptions. Assuming quite short reads of 300bp and requiring 55bp of overlap to connect reads, does not significantly degrade effective coverage as shown in the lower of the two curves in the figure.

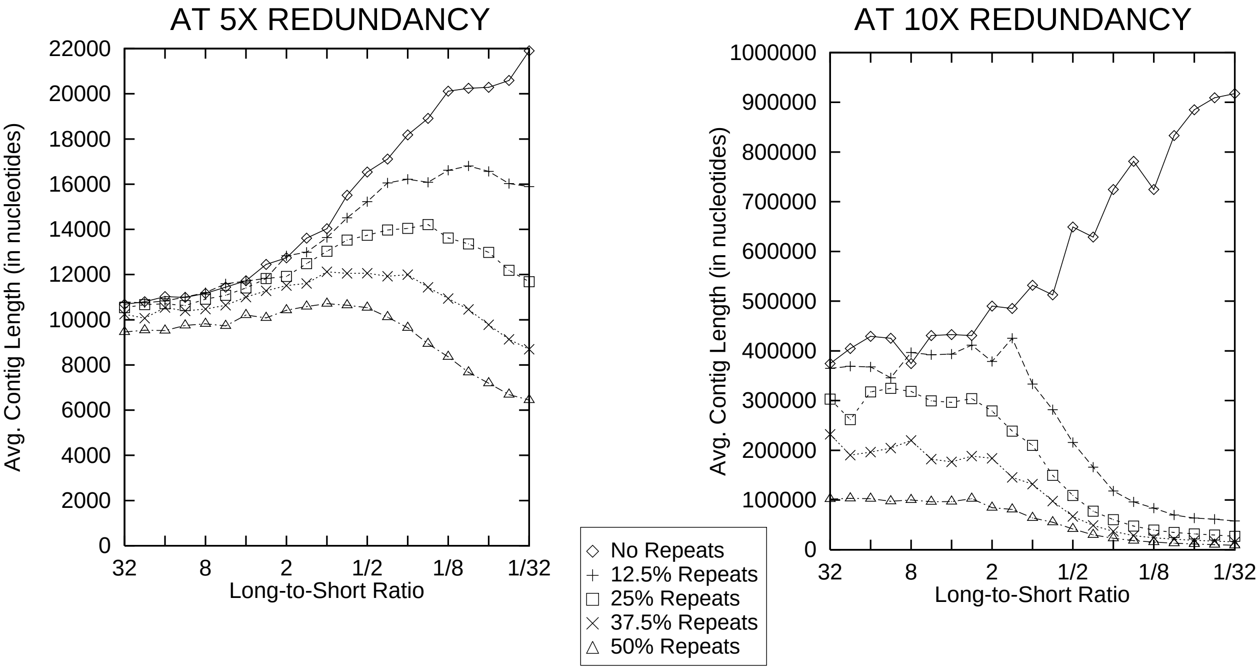

Figure 5: Average contig length as a function of long/short mix, repetitiveness, and sequencing redundancy.

The total number of contigs and gaps in a whole genome data set depends on the average length of a contig which in turn depends on the mix of long to short inserts and the repetitiveness of the genome. The plots in Figure 5 depict curves for the average contig length as a function of `LS_RATIO`, one curve for a different degree of repetitiveness of the underlying genome. Figure 5a shows the situation assuming a 5x sequencing redundancy, and Figure 5b is for 10x sequencing. Assuming no repeats in the genome, short inserts are effectively 700bp reads and long inserts provide 400bp reads, the connection between pairs being of effectively no value in terms of contig building. Thus a greater ratio of shorts is produces longer contigs. However as soon as one begins to place repeats into the genome, the LINE elements become a barrier which short inserts cannot cross. Thus when all inserts are short, the average contig length quickly becomes clamped to the average length between LINE elements. At the other end of the curves, where all inserts are long, there is a slight degradation but it is limited only by the number of inserts that are rendered useless. Note that in Figure 5a where sequencing redundancy is 5x, a mix of shorts increases contig length up to about a 1 to 2 ratio, but in Figure 5b, where sequencing is 10x, there is only a steady loss as shorts are introduced into the mix, with a precipitous decline in contig length as the ratio becomes less than 1 to 1. One would conclude that the mix ratio should therefore be kept at least this high.

The other aspect of whole genome sequencing that we investigated via simulation, was the nature of the scaffolds linking contigs together. We found that in almost all cases, that at the default settings and for a ‘‘super-chromosome’’ of 500Mbp, that (1) there was one scaffold that spanned almost the entire chromosome, (2) there were 2-5 small scaffolds of 2-5 contigs near the ends of the chromosome, and (3) there were 10-20 isolated, and small contigs of one or two reads nestled between the contigs of the super-scaffold. Thus the scaffold information linked together is very strong, effectively giving a global linking of all contigs.

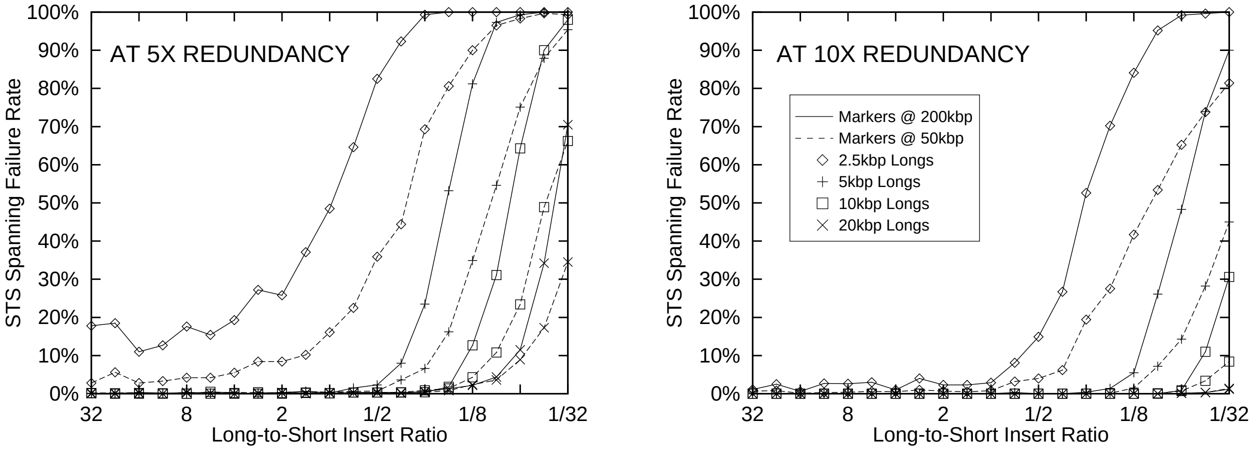

Figure 6: STS spanning failure rate as a function of long/short mix, long insert length, inter-marker spacing, and sequencing redundancy.

Recall that to assemble a whole genome data set, we will be assuming the availability of an STS map of some average resolution, hopefully on the order of 100Kbp between each marker. The aspect of scaffold connectivity that is thus the most relevant is the frequency with which a scaffold exists that links together a given pair of markers in such a map. To this end, the simulator was augmented to generate pairs of markers of length `MARKER_LEN` at a distance `MARKER_SPACING` apart. The location of the markers was chosen uniformly subject to the additional criterion that the markers were not in a repetitive element. The default length of markers was 300bp and for data point in Figure 6 was generated with 1000 trials on marker pairs. Each plot displays curves showing the effect of the long-to-short insert mix, the length of the long inserts, and the spacing between the markers. What is plotted is the percentage of the trials for which a scaffold containing both markers failed to exist, i.e., the ‘‘STS spanning failure rate’’. Figure 6a shows a series of curves for 5x sequencing redundancy and Figure 6b for 10x sequencing. The results shown are very encouraging: for long inserts of 5Kbp or more and a long-to-short ratio of 1 or more, a spanning scaffold exists over 99% of the time for redundancies as low as 5x. One should further consider that these plots assume that 5% of the genome between the markers is part of long repeats that are barriers to connectivity.

2.3 Inter-Marker Assembly:

We now turn to an examination of the problem of determining the sequence between two markers, *A* and *B*, given a whole-genome data set. First consider the problem of finding a scaffold that connects the two markers. Imagine a graph in which each vertex represents an insert (not a read) and where there is an oriented edge between two vertices if their inserts have overlapping reads. Let O_A be the set of inserts whose reads have a significant overlap with marker *A*, and let O_B be the corresponding set for the marker *B*. It is easy to see that a scaffold between *A* and *B* must corresponds to a path from a vertex in O_A to a vertex in O_B, i.e., our problem reduces to a simple graph connectivity question. The difficulty is that while a ‘‘true’’ scaffold must correspond to such a path, there may be connecting paths that involve false overlaps and hence do not model a valid spanning scaffold.

In a redundancy $\overline{c}$ data set, one would expect that each read will on average have true overlaps with $2(\overline{c}-1)$ other reads. However, a read that contains part of a repetitive element will have many more overlaps due to the repeat overlaps induced by all the other copies of the repetitive element. In terms of the graph, reads that do not contain overlaps will typically have degree $O(\overline{c})$ whereas those containing repeats will have degree $O(\overline{c}\,n)$ where *n* is the number of copies of the repeat in the genome. While there is statistical fluctuation in the degrees of vertices, note that vertices containing high copy repeats such as Alu’s will easily be distinguished by their enormously large edge degree. But an even more useful observation is that vertices with degrees less than $2\overline{c}$ very likely do not contain repeats and hence all their edges/overlaps are very likely to be true.

It now follows from the simulation results of the previous section that a path from O_A to O_B using predominantly low degree vertices is very likely to exist. Thus while we have yet to simulate the following, we venture that the following algorithm will find a minimal scaffold with high probability. The algorithm first finds the vertices in O_A and O_B and then begins a depth first search traversal from these vertices towards those in the opposing set in parallel. Because the edges are oriented the algorithm knows in which direction it is extending. If extending from a point estimated to be less than half way between the two markers, then the algorithm explores long inserts before short inserts, otherwise it gives preference to low-degree vertices over high-degree vertices. The algorithm terminates when paths from opposite markers meet.

From our prior simulation experience we know that the average number of inserts in a spanning path is on the order of the marker spacing divided by the length of long inserts and that favoring long inserts up to the half-way point significantly reduces the number of edges that need to be explored. Because high degree vertices are avoided it will almost always be the case that the spanning scaffold found will not involve any false overlaps. The occasional exception will be for inter-marker stretches containing several copies of a low-copy number and very large repeat (such as a gene). Even in these cases, the repeats will be skirted unless they are tandem or chance works against the algorithm. However, provided that insert sizes are know to within plus or minus 50% this situation will be detected because the path found will be too short. Further evidence for this situation will be the presence of one or more distinct loops in the paths traversed, and possible dead-end paths induced by copies of the repeat in other parts of the genome. Figure 7 provides an illustration.

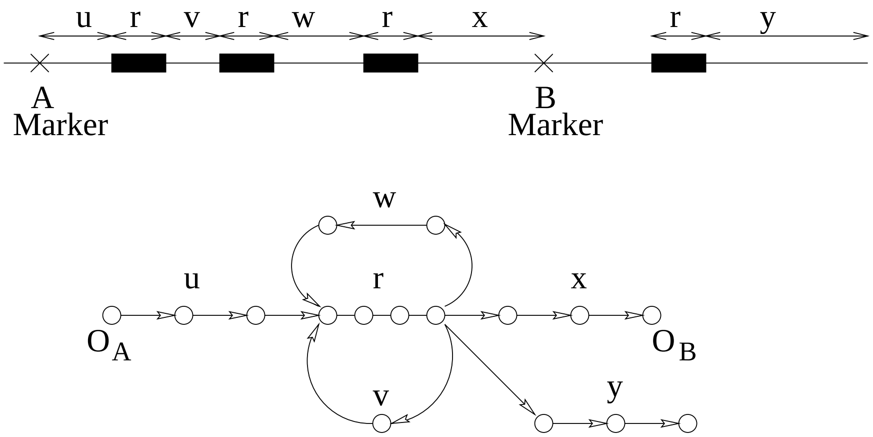

Figure 7: Illustration of inter-marker path structure as a function of the inter-marker sequence.

Given a minimal and true scaffold between two markers the inter-marker assembly problem then becomes like any other shotgun assembly problem save that pulling in the relevant pieces involves much more time intensive overlap comparisons than in standard problems. We estimate that with current high-performance workstatsion, one inter-marker assembly between markers at 100Kbp spacing and a 10x data set over the human genome can be solved in one day. Ultimately, how well or easily the inter-marker assembly problem can be solved depends upon how tightly the lengths of inserts are known and the degree to which dual end reads are asserted to be mates but are actually not (because of chimerism or data tracking errors). It is at this time an open question to ascertain what can be expected but our preliminary experience suggests a positive result.

3. DISCUSSION

The map-based whole genome assembly problem may be succinctly stated as follows. One is given a 5-10x shotgun data set of end-sequenced inserts and an STS map of the target genome. The problem is to determine, for each adjacent pair of markers, the sequence between the markers. There are two things to note immediately. First, is that the proposition is applicable to not only whole genomes but to any assayable target. One might consider sequencing a large target by simultaneously collecting shotgun data while developing a 100Kbp resolution STS map. Second, is that the computational problem is divided into a collection of independent inter-marker assembly problems, one for each pair of adjacent markers. Thus a very large computation is nicely broken into manageable subproblems that may be solved in parallel.

From the simulation results it follows that, with a suitable choice of redundancy, long insert length, and long-to-short insert mix, one will find a scaffold spanning a pair of markers over 99% of the time. Moreover, one can then assemble the contigs of this scaffold between the marker. It follows from these simulation results that one can expect a number of contigs and average gap length between these contigs that is *identical* to that which would have been obtained had one shotgun sequenced the stretch between the two markers. The only drawback is that the extreme order statistic for gap length is larger than in clone-by-clone approaches so there will be a small percentage of cases where gaps can be PCR-filled in the clone-by-clone data sets but not in the whole genome case. Otherwise the amount of information is equivalent. At the default setting of the parameters a scaffold will exist 99.8% of the time, and in such cases 99.6% of the sequence between the markers will be delivered, implying that 99.4% of the genome will result from such a data set in regions that are not wildly repetitive (e.g. centromeres and telomeres).

The principal appeal of map-based whole genome sequencing is its simplicity. There is no physical library to construct or to maintain, greatly simplifying the number of steps, and, more importantly from the point of view of automation, the number of *types of steps* involved. There is no need to track reads, the only data organization required is list of the reads and their long insert mates. Clone-by-clone approaches such as whole genome OSS will involve resequencing 5-10% the genome as this much overlap is estimated in a $\bar{m} = 20$ minimum tiling set. Whole genome sequencing does not suffer this waste of effort. It is also possible to immediately derive comprehensive polymorphism information by collecting inserts from several different individuals. The clone-by-clone approaches would require completely sequencing each individual to obtain the same information.

On the other hand, there are drawbacks to map-based shotgun sequencing. First, the STS spanning failure rate does not become small until sequencing redundancy reaches 4x. Thus one must perform 40-80% of all sequencing, before one can begin to actually deliver sequence information. While we have given an indication that false repeat overlaps can effectively be inferred and that inter-marker assembly is very likely computationally feasible, we have not definitively demonstrated this. Therefore, the risk of collecting so much data without the certainty of successful prior experience is daunting.

For the future, we plan on further simulation aimed at exploring the reliability/feasibility of inter-marker assembly. In a first phase we will do so by simulations of the type presented. If these reveal a positive outcome, then we will proceed with the more expensive trials of building an inter-marker assembler and running it on data sets obtained by taking the largest available stretches of human DNA and building a simulated whole genome data set from an extrapolation of said sequence. In a final test-of-feasibility stage, a real data set could be generated for a 1-5Mbp segment of a genome.

REFERENCES

1. Sanger, F., Coulson, A.R., Hong, G.F., Hill, D.F., and Peterson, G.B. 1982 Nucleotide sequence of bacteriophage λ DNA. *J. Mol. Biol.* 162:729.
2. Sanger, F., Nicklen, S. and Coulson, A.R. 1977. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA* 74:5463.
3. Peltola, H., Sӧderlund, H., and Ukkonen, E. 1984. SEQUAID: a DNA sequence assembly program based on a mathematical model. *Nucleic Acids Research* 12:307-321.
4. Kececioglu, J. and Myers, E. 1995. Exact and Approximate Algorithms for the Sequence Reconstruction Problem. *Algorithmica* 13:7.
5. Myers, E. 1995. Toward Simplifying and Accurately Formulating Fragment Assembly. *J. of Computational Biology* 2:275.
6. L. Rowen, and B.F. Koop. Zen and the art of large-scale genomic sequencing. *Automated DNA Sequencing and Analysis* (M.D. Adams, C. Fields, & J.C. Venter, eds.) Academic Press (London, 1994), 167-174.
7. Bell, G.I. 1992. Roles of repetitive sequences. *Computers Chem.* 16:135.
8. Iris, F.J.M. 1994. Optimized methods for large-scale shotgun sequencing in Alu-rich genomic regions. *Automated DNA Sequencing and Analysis* (M.D. Adams, C. Fields, J.C. Venter, eds.) Academic Press, London, 199-210.
9. Lander, E.S. and Waterman, M.S. 1988. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* 2:231.
10. Wilson, R.K., Chen, C., and Hood, L.E. 1990. *BioTechniques* 8:184.
11. Edwards, A., and Caskey, C.T. 1991. *Methods: Companion Methods Enzymol.* 3:41.
12. Berg, C.M., Wang, G., Strausbaugh, L.D., and Berg, D.E. 1993. *Methods Enzymology* 218:279.
13. Myers, E Advances in Sequence Assembly. *Automated DNA Sequencing and Analysis* (M.D. Adams, C. Fields, & J.C. Venter, eds.) Academic Press (London, 1994), 231-238.
14. Larson, S., Jain, M., Anson, E., and Myers, E. An interface for a fragment assembly kernel. *Tech. Rep. TR96-04*, Dept. of Computer Science, U. of Arizona, Tucson, 85721.
15. Roach, J.C., Boysen, C., Wang, K., and Hood, L. 1995. Pairwise end sequencing: A unified approach to genomic mapping and sequencing. *Genomics* 26:345.
16. Chen, E.Y., Schlessinger, D., and Kere, J. 1993. Ordered shotgun sequencing, a strategy for integrated mapping and sequencing of YAC clones. *Genomics* 17:651.
17. Hudson, T.J. et al. 1995. An STS-based map of the human genome. *Science* 270: 1945-1954.
18. Downey, P.J., and Maier, R.S. 1988. Logarithmic moment bounds for extreme order statistics. Technical Report TR 88-32, Dept. of Computer Science, University of Arizona, Tucson, AZ 85721-0077.