Toward Simplifying and Accurately Formulating Fragment Assembly

Eugene W. Myers*

Abstract. The fragment assembly problem is that of reconstructing a DNA sequence from a collection of randomly sampled fragments. Traditionally the objective of this problem has been to produce the shortest string that contains all the fragments as substrings, but in the case of repetitive target sequences this objective produces answers that are overcompressed. In this paper, the problem is reformulated as one of finding a maximum-likelihood reconstruction with respect to the 2-sided Kolmogorov-Smirnov statistic, and it is argued that this is a better formulation of the problem. Next the fragment assembly problem is recast in graph-theoretic terms as one of finding a non-cyclic subgraph with certain properties and the objectives of being shortest or maximally-likely are also recast in this framework. Finally, a series of graph reduction transformations are given that dramatically reduce the size of the graph to be explored in practical instances of the problem. This reduction is very important as the underlying problems are NP-hard. In practice, the transformed problems are so small that simple branch-and-bound algorithms successfully solve them, thus permitting auxiliary experimental information to be taken into account in the form of overlap, orientation, and distance constraints.

Keywords: shotgun DNA sequencing, fragment assembly, sequence reconstruction.

1 Introduction

In the mid 1970's experimental procedures were devised [SNC-77, MaG-77] that produced a ladder-like pattern on a permeable gel, permitting one to list, in order, the first 200-300 nucleotides of a DNA strand. These gel-electrophoretic procedures have since improved so that with great care one can now interpret upwards of the first 1000 nucleotides of a sample, but the length of such a "*read*" will always be limited by the resolution of the images. In order to determine the sequence of much longer segments of DNA, Sanger et al. [SCH-82] devised what is known today as the *shotgun* strategy: sample fragments as randomly as possible from the target sequence and then read as much as possible of the initial sequence of each of these fragments via gel-electrophoresis. In such an experiment one should clearly sample fragments whose length is longer than the maximum expected length of a read. Provided that enough fragments are sequenced and their sampling is sufficiently random across the target, one then expects to be able to determine the target by finding sequence overlaps

* Dept. of Computer Science, University of Arizona, Tucson, AZ 85721 (e-mail: gene@cs.arizona.edu). Partially supported by NLM grant LM-04960 and DOE grant DE-FG05-91ER61132.

among the reads of fragments that were sampled from overlapping stretches. The essential and most difficult aspect of this *computational* problem is to determine a layout or arrangement of the fragment reads that is most consistent with the overlaps found between them. The difficulty of this already NP-hard problem is compounded by the fact that (1) there is a low level of error, typically 1 to 5 per cent, in the experimentally determined fragment reads, (2) DNA is a duplex and so the reported fragment reads could be from either strand of the duplex, and (3) if insufficient sampling has occurred or there is a strong biological bias in the sampling, then not all of the target will necessarily be represented in the fragment population.

This last problem becomes progressively more acute just on the basis of statistical considerations [LaW-88] as the ratio of the length of the target to the length of the reads increases. While read length has increased some in the last decade, the desire of experimentalists to sequence ever larger stretches has lead to ratios where one does not usually reach complete coverage after having sequenced an amount of data equal to four to six times that of the target. To circumvent this, experimentalists then use more expensive directed sampling methods to collect fragments that fill the gaps in the target. Other approaches that ameliorate this coverage problem include, collecting larger fragments and reading both ends of each fragment (effectively giving a read of twice the length), or selectively collecting longer reads of fragments adjacent to the gaps using better (but more expensive) technology. There is the possibility of selecting all the fragments in a directed fashion so that the problems of sampling and determining a layout disappear. But as yet these methods incur greater cost in terms of reagents and time, and have not been as amenable to the design of assembly-line procedures. Thus the predominant mode of large scale sequencing as of the present time is what we term hybrid shotgun sequencing: a combination of predominantly randomly sampled data with some additional directed components.

Rather than design different algorithms for each hybrid strategy, we find it preferable to think of and design algorithms that are capable of solving a "pure" shotgun problem subject to a collection of overlap, orientation, and distance constraints that model the additional information provided by the directed components of the strategy. We do not address such a "shotgun-with-constraints" problem in this paper, but we raise it here because it points out that simpler or better algorithms for the "pure" problem are required if there is to be any hope of solving these more difficult constraint problems that are even further complicated by the fact that typically 10% of the constraints are in fact erroneous. The simplification strategy presented in the last section of this paper has such potential.

Besides hybrid strategies, the other major consequence of recent trends in the scale and scope of DNA sequencing activities is that it is now quite commonplace for the target sequences to contain repetitive elements. For example, in the T-cell receptor locus of humans there is a 5-fold repeat of a trypsinogen gene that is 4kb (kilo-bases) long and varies 5-10% between copies. Three of these were close enough together that they appeared in a single shotgun-sequenced cosmid target[RoH-94]. 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 300, do not share this feature but still are problematic as they can constitute up to 50-60% of the target sequence [Bel-92, Iri-94]. 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 [Bel-92]. The problem of repeats has only become acute in recent years because researchers are beginning to sequence higher organisms that have a greater tendency towards such sequence, and because with increasing target length, there is greater chance of a large scale repeat being within the target. The work in this paper specifically addresses this issue where in the next section it formulates the objective of the fragment assembly problem in a way that is correct for repetitive DNA, and further develops this criterion into the simplification strategy presented in its final section.

Proceeding formally, this paper considers the "pure" version of shotgun sequencing formulated as follows. The *fragment assembly problem* is to determine the sequence of bases in an unknown target DNA duplex S of length G. Conceptually, S is a string over the four letter alphabet A,C,G,T. In shotgun sequencing, an investigator samples the target S, say F times, obtaining in each sampling a *fragment read* f_i of average length $\overline{L}$ that is a substring of S or its dyadic complement, and that has been perturbed by the introduction of $\varepsilon|f_i|$-or-fewer differences. The fraction $\varepsilon \in [0,1]$ models the *maximum* error rate of the sequencing process and is typically about 5-10%. The dyadic complement f^c of a sequence f represents the strand complementary to f when it is in duplex DNA. Formally, $(a_1 a_2 ... a_n)^c = wc(a_n)...wc(a_2)wc(a_1)$ where $wc(A) = T$, $wc(T) = A$, $wc(C) = G$, and $wc(G) = C$. We let $N = F\overline{L}$ denote the total amount of sequence data collected by the investigator, and let $\overline{c} = N/G$ denote the average *coverage* or sampling frequency of each base. With current technology the average fragment read length $\overline{L}$ is in the range 300-500 and investigators are shotgun sequencing strings S of length up to 50,000. A typical coverage for such sequencing projects is $\overline{c} = 6$, requiring that $F = \overline{c}G/\overline{L} = 1000$ fragments be sampled if one conservatively assumes $\overline{L} = 300$.

2 A New Formulation of Fragment Assembly

Traditionally, the fragment assembly problem has been phrased as one of finding a shortest common superstring (SCS) of the fragment reads within error rate ε, i.e., a shortest string R such that for every fragment read f_i, there is a substring $R[sp_i, ep_i]$ that is not greater than $\varepsilon|f_i|$ differences from either f_i or ${f_i}^c$. But for problems involving repetitive DNA sequence this criterion clearly over-compresses repeats that are larger than $\overline{L}$ and so is not a correct formulation either in theory or practice. Figure 1 gives an example of a target for which such an over-compression occurs. While it is certainly true that knowing an SCS implies the original target DNA duplex is known in the sense of learning theory [Li-90], the plain fact is that in practice SCS-based algorithms do not correctly handle repetitive sequence (e.g. [Kec-91]). Note that R may be the concatenation of disjoint "contigs" of mutually overlapping fragments. In Figure 1 this is due to incorrect assembly, but more generally, it is due to the fact there are gaps in the sampling coverage of the original target.

In Figure 1, the fact that the solution has over-compressed the repeated part is clear by the unusually high coverage of that part of the solution. This suggests that we might avoid this pitfall by formulating the fragment assembly problem in terms of finding the sequence that maximizes the likelihood of the hypothesis that its fragments were sampled over the length of the target with a given distribution.

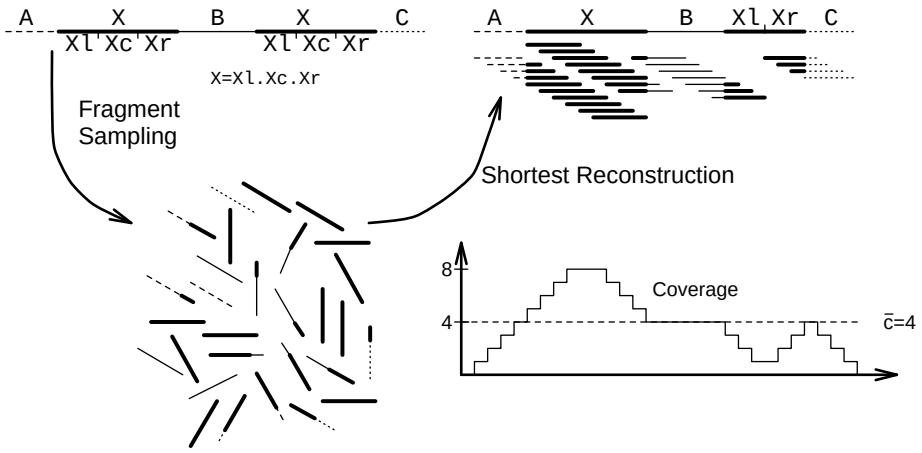

Fig. 1. An over-compressed SCS-based assembly.

The probability density function of the Kolmogorov-Smirnov test statistic for the goodness-of-fit between an observed sample and a hypothesized source distribution gives us a suitable likelihood function. We will assume later in this paper that the source distribution is the uniform distribution, but this is not essential to our formulation as the Kolmogorov-Smirnov test statistic is distribution-free. We choose the uniform distribution because it is indeed the desire of experimentalists to approximate this assumption with sequence-independent methods of fragmentation such as sonication.

Proceeding formally, let a particular solution be described by a *reconstruction string* R, and a *layout* consisting of F pairs of integers, $(s_i, e_i)_{i\in[1,F]}$, where $1 \leq s_i, e_i \leq |R|$. The i^{th} pair of the layout indicates that f_i is a perturbed copy of the substring $R[s_i, e_i]$ if $s_i \leq e_i$, or of $R[e_i, s_i]^c$ otherwise. Thus the order of s_i and e_i encode the *orientation* of the fragment read in the layout, that is, whether f_i was sampled from R or its complement strand. We say that the *start-point* sp_i of read f_i in the layout is $min(s_i, e_i)$ and its *end-point* ep_i is $max(s_i, e_i)$. In order to be ε-*valid* a layout must satisfy the following two properties: (a) each read f_i can be aligned to its assigned substring with not more than $\varepsilon|f_i|$ differences, and (b) every symbol of R must be covered by some read, i.e. $\cup_i[sp_i, ep_i] = [1, |R|]$. Our potential solution space is thus the set of all ε-layouts, and our problem is to pick one as "best".

Observe that given a layout, we have an observed distribution $D_{obs}(x)$ of fragment read start points:

$$D_{obs}(x) = |\{f_i : sp_i < x\}|/F$$

that is the proportion of reads starting before position x. We let the domain of the distribution be $1 \leq x \leq G' + 1$ where $G' = G - \overline{L}$ because a fragment of length $\overline{L}$ cannot begin at a position greater than $G'+1$[2]. Given a known continuous source distribution $D_{src}(x)$ for the sampling process, the 2-sided Kolmogorov-Smirnov statistic [Kol-33, Smi-41] is the maximum deviation $\delta = max_{1\leq x\leq G'+1}|D_{obs}(x) - D_{src}(x)|$ between the source and observed distributions. The probability distribution $Pr_{\text{2K-S}}(\delta)$

[2] This *approximation* effectively dismisses boundary effects. It suffices in practice as $\overline{L}$ is 1% or less of G.

$= Pr(x \leq \delta)$ of this statistic is independent of that of D_{src} and the following analytic formula [Kol-33, Bir-52] permits its computation for any F and δ:

$$Pr_{\text{2K-S}}(\delta) = \frac{F!}{F^F} H^F[\lfloor F\delta \rfloor + 1, \lfloor F\delta \rfloor + 1]$$

where $H^F[i,j]$ is the entry in row i and column j of the F^{th} power of the $p \times p$ matrix:

$$H = \begin{bmatrix} 1-\kappa & 1 & 0 & 0 & \cdots & 0 \\ \frac{1-\kappa^2}{2!} & 1 & 1 & 0 & \cdots & 0 \\ \frac{1-\kappa^3}{3!} & \frac{1}{2!} & 1 & 1 & \cdots & 0 \\ \frac{1-\kappa^4}{4!} & \frac{1}{3!} & \frac{1}{2!} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1-\kappa^{p-1}}{(p-1)!} & \frac{1}{(p-2)!} & \frac{1}{(p-3)!} & \frac{1}{(p-4)!} & \cdots & 1 \\ \frac{1-2\kappa^p+max((2\kappa-1),0)^p}{p!} & \frac{1-\kappa^{p-1}}{(p-1)!} & \frac{1-\kappa^{p-2}}{(p-2)!} & \frac{1-\kappa^{p-3}}{(p-3)!} & \cdots & 1-\kappa \end{bmatrix}$$

where $p = 2\lfloor F\delta \rfloor + 1$ and $\kappa = 1 - (F\delta - \lfloor F\delta \rfloor)$.

One way to define our objective in maximum-likelihood terms would be to score each ε-layout according to the value of the probability density function $dPr_{\text{2K-S}}(\delta)/d\delta$ which is unimodal with a peak at $\delta_{max} > 0$. However, such an approach would be computationally cumbersome due to the difficulty of computing this likelihood function at a given δ. Keeping in mind that we are not testing the hypothesis that the fragments were sampled with distribution D_{src}, but rather are seeking ε-layouts that conform to this *fact*, it would be rare to observe layouts whose δ is less than δ_{max}. Thus we take the liberty of simplifying matters considerably by taking as our objective the minimization of δ. In essence, we are eliminating layouts with large δ that are unlikely to satisfy the null-hypothesis that $\{sp_i\}$ was sampled from D_{src}. Indeed, the algorithms that follow are capable of generating all layouts whose goodness-of-fit is within a certain level of significance.

A subtlety not yet addressed is that a layout and its reconstruction do not necessarily cover all of the original target, i.e., it may be that $|R|$ is a bit less than G because with some positive probability fragments were not sampled from either end of the target. Alternatively, while sp_1, the start-point of the first fragment, is 1 from the view of R, it is not necessarily the first position in the target. In the absence of any other information, we assume that we are free to arbitrarily place R with respect to the target and thus shift D_{obs} and D_{src} with respect to one another in order to minimize the deviation. That is, we use as our optimization objective the *relative* deviation

$$\delta = min_\alpha \ (\ max_{1 \leq x \leq G'+1} |D_{obs}(x) - D_{src}(x-\alpha)| \ )$$

While this invalidates the use of δ as a test statistic, it does not invalidate its use here as an objective function. Figure 2 illustrates these concepts for the case where D_{src} is the uniform distribution. At this point, we are now ready to formulate the fragment assembly problem as follows:

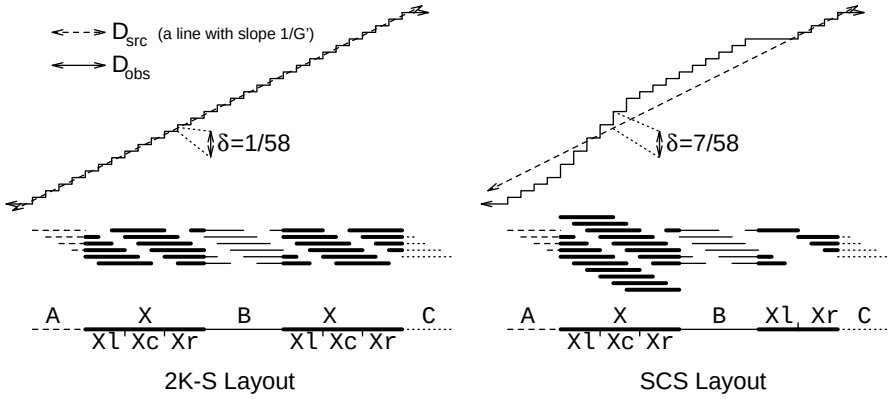

Fig. 2. Deviation between source and observed start point distributions.

Fragment Assembly: Given fragment reads f_i and maximum error rate $\varepsilon \in [0, 1]$ find a reconstruction R and ε-valid layout of the reads whose observed distribution of fragment read start points, D_{obs}, has the the minimum relative deviation from D_{src}.

Note carefully that the problem is phrased as finding the maximum likelihood *ε-valid* layout. Thus we are still constraining the space of solutions to "noisy" superstrings of the fragment reads, i.e., reads can only overlap in a proposed layout if they agree within ε-or-less errors along the length of their overlap. Indeed, later in the paper, our likelihood-based objective is only used to select among the few competing layouts that remain after all "unique" overlaps are utilized. The point is that rather than evaluating layouts based on the length of their reconstruction, we seek the ones that are most consistent with the fact that the start points of fragments were chosen across the length of the target with distribution D_{src}. Indeed, it is our initial experience that our new criterion only produces different results from SCS in repetitive regions which is exactly where prior work fails. The analytic formula for $Pr_{\text{2K-S}}$ confirms that our criterion is not limited to asymptotically large sample sizes, but applies to a sample of *any finite size* F. Finally, it should also be emphasized that D_{src} can be *any distribution*; in this sense the distribution, $Pr_{\text{2K-S}}(\delta)$, of the Kolmogorov-Smirnov statistic δ is distribution-free. Thus if one has more refined information about the *a priori* distribution of fragments than one can use this knowledge in our formulation.

In summary, viewing the fragment assembly problem as a "noisy" shortest common superstring problem is an appeal to parsimony: one seeks the explanation that is shortest. While parsimony is a much used criterion in computational biology, it fails regularly in practice on fragment assembly problems involving repetitive sequence. Our alternative formulation of finding a maximum-likelihood solution is also well-founded theoretically and in preliminary trials appears to produce better results in practice[3].

[3] One could imagine other goodness-of-fit measures such as the maximum deviation from average coverage $\overline{c}$, or the area of deviation between D_{src} and D_{obs}, or by using a χ^2-

3 A Graph-Theoretic View of Fragment Assembly

A common approach to fragment assembly, advocated by several authors [PSU-84, Kec-91, Hua-92], divides the problem, both conceptually and computationally, into three phases: *overlap*, *layout*, and *consensus*. In the overlap phase, one compares every fragment read against every other read (in both orientations) in search of approximate overlaps between reads. These overlaps are recorded and capture all of the possible relationships between the fragment reads. The layout phase then selects a subset of these pairwise overlaps that determines the location of every fragment read with respect to every other. In essence, the layout phase determines the pairs (s_i, e_i) discussed in the previous section to within an accuracy that is dependent on ε. Finally, the consensus phase forms a consensus-measure multi-alignment in all regions where the coverage is two or greater in order to select a consensus character for each position resulting in the ultimate reconstruction R. Note that it is only until R has been computed in the last phase that the exact values for the pairs (s_i, e_i) of a layout are known.

The focus of this paper is on the middle, layout phase. Consequently, we must introduce the necessary preliminaries for the overlap phase, but will not discuss consensus further. For more details on the entire process see [KeM-95]. In our conception, the overlap phase is an "all-against-all" variation of sequence comparison that compares every fragment read against every other read *and* its dyadic complement in search of approximate overlaps between them that are statistically significant at a level settable by the user but usually with a probability of at most 1-in-10^5 of occurring by chance (see [Mye-86, Kec-91, KeM-95]). Given a maximum error rate ε, the only alignments between overlapped portions of two reads A and B that need be detected involve not more than $k = \varepsilon(|A| + |B|)$ differences.

We view the overlap phase as producing an *overlap graph* $\mathcal{G}$ that models all the approximate overlaps reported by an algorithm that performs the comparisons described in the previous paragraph. Each fragment read is modeled as a vertex and each reported overlap as an edge. The overlap graph may be a multi-graph in that there may be more than one edge between a pair of reads A and B. This may result from there being more than one significant locally optimal overlap between them, but more often it is because there is a significant overlap between A and B, and another between A and B^c. In order to accurately model layouts, the overlap graph must model more than the two reads $\pi.A$ and $\pi.B$ involved in an overlap π and so must be more than an undirected multi-graph. The convention is adopted that an overlap is always described from the perspective of $\pi.A$ being in the *forward* orientation (i.e., uncomplemented) and that of $\pi.B$ being variable. The complete specification of an overlap π is achieved by specifying (a) the substrings $\pi.A[\pi.s_A, \pi.e_A]$ and $\pi.B[\pi.s_B, \pi.e_B]$ of each read involved in the overlap, and (b) a list $\pi.\Delta$ of the positions of the unaligned symbols of each read in the alignment of the overlap. Since π is described from the perspective of $\pi.A$, it is always the case that $\pi.s_A \leq \pi.e_A$, but as when specifying a layout, $\pi.s_B > \pi.e_B$ indicates that

based hypothesis test. We choose Kolmogorov-Smirnov because it is easy to compute, consistent, and distribution-free. An empirical test and comparison of alternatives is beyond the scope of this paper. Our aim here is to rigorously propose *some* alternative to parsimony as a layout optimization criterion that accounts for repetitive sequence.

the overlap is between $\pi.A[\pi.s_A, \pi.e_A]$ and $\pi.B[\pi.e_B, \pi.s_B]^c$. $\pi.\Delta$ is often called a Δ-encoding of an alignment and while there are other ways to encode an alignment, the Δ-encoding is the most space efficient in our context as ε is rarely more than 10%. While $\pi.\Delta$ is not needed by the layout phase, the orientation of the B-read and the substrings involved in an overlap are important to the layout phase and the range of possibilities is partitioned into four categories of edges shown in Figure 3.

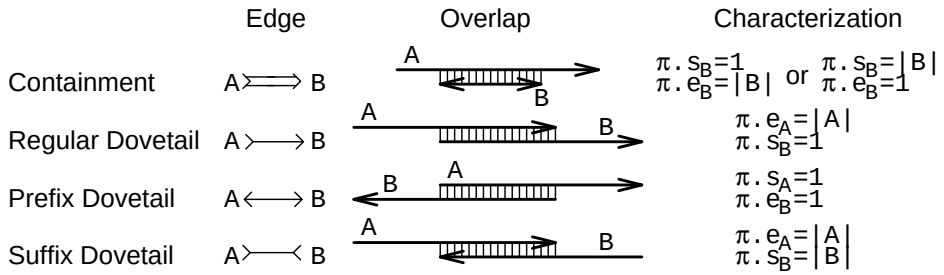

Fig. 3. Taxonomy of Overlap Types.

A *containment* edge models the situation where all of $\pi.B$ is aligned to a substring of $\pi.A$, and is denoted by a directed, double-line edge from $\pi.A$ to $\pi.B$. All other edges are *dovetail* edges where a proper prefix or suffix of one read is aligned to a proper prefix or suffix of the other. Each dovetail edge is denoted by a bi-directed, single-line edge. The edge is bi-directed in that there is an arrowhead at each end of the edge and the direction of each arrowhead is a significant part of the model. When $\pi.B$ is in the same orientation as $\pi.A$ we have a *regular* dovetail edge where a suffix of $\pi.A$ overlaps a prefix of $\pi.B$ and the arrowhead at $\pi.A$ is directed out and the arrowhead at $\pi.B$ is directed in. When $\pi.B$ is in the opposite orientation the overlap can involve either a prefix or a suffix of $\pi.B$. For a *prefix* dovetail edge, the arrowheads are directed into both $\pi.B$ and $\pi.A$. For a *suffix* dovetail edge, both arrowheads are directed outward. We leave it to the reader to verify that every overlap falls into exactly one of these categories when given the freedom to choose which read is the A-read. The essential property of a dovetail edge is that when one of its arrowheads is directed into a read the prefix of the read is in the overlap, and when directed out, the suffix of the read is in the overlap.

To sum up to this point, the result of the overlap phase is the construction of an overlap graph $\mathcal{G}$ that is a bi-directed, multi-graph of containment and dovetail edges. Each edge/overlap is further annotated with the exact substrings involved in the overlap's alignment and its Δ-encoding. The question now at hand is how to describe the class of ε-valid layouts in terms of this graph-theoretic construction. To this end let the *dovetail in-degree*, $dove_{in}(f)$, of a vertex f be the number of *dovetail* arrowheads directed into it, and let $cont_{in}(f)$, its *containment in-degree* be the number of *containment* arrowheads directed into it. Similarly, define the outdegrees, $dove_{out}(f)$ and $cont_{out}(f)$.

Lemma 1 [Kec-91]: Every ε-valid layout is modeled by a non-cyclic subgraph of $\mathcal{G}$ that satisfies (a) $dove_{in}(f)$, $dove_{out}(f)$, $cont_{in}(f) \leq 1$, and (b) $cont_{in}(f) = 1 \Longrightarrow dove_{in}(f) = dove_{out}(f) = 0$.

Proof: As a first step, sequentially remove contained fragment reads from the layout and place in the initially empty model a containment edge that is from a read still in the layout to the read being removed. Such a containment edge must exist in $\mathcal{G}$ as the layout is ε-valid. The result is a forest of containment trees and a layout in which all remaining reads have only dovetail overlaps between them. Note that the layout may consist of different connected components (modeling the fact that the reads do not cover the original target) called *contigs*. Certainly the reads remaining in each contig can be ordered from left to right according to their start points. Add to the model the dovetail edges between consecutive reads in this ordering and by the construction of arrowhead directions (i.e., arrowhead in iff a prefix, arrowhead out iff a suffix), we have a simple path of dovetail edges for each contig satisfying the dovetail-degree constraints. Again, all these dovetail edges must exist in $\mathcal{G}$ (except when the reads don't overlap in the layout) as the layout is ε-valid. Finally, property (b) is true as the contained reads were removed from consideration in the first step. □

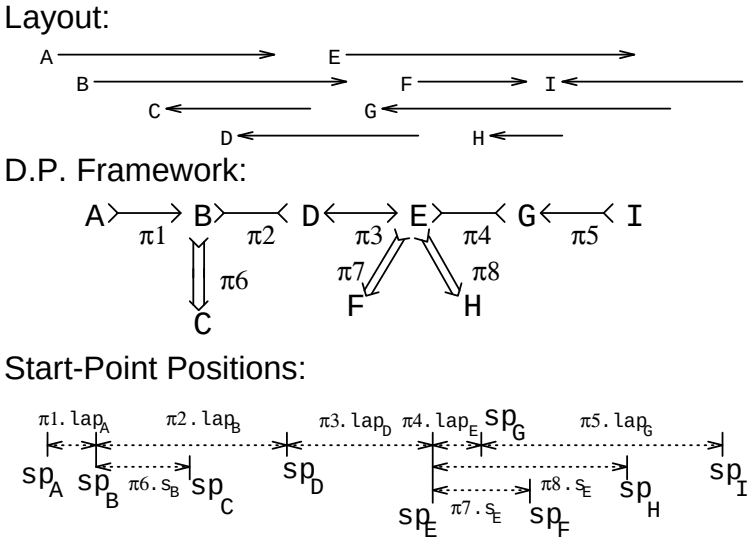

Fig. 4. D.P. Framework for a Layout and Its Start-Points.

We call a subgraph satisfying the conclusion of Lemma 1 a *dovetail path framework* or more briefly a d.p. framework, because it consists of one or more dovetail paths (subject to in/out-degree constraints) that form frames off of which hang trees of containment edges. In what follows, we will generally just focus on a given connected component or contig, with the understanding that the discussion applies to each con-

tig of the d.p. framework. Lemma 1 guarantees that in order to find a reconstruction and accompanying ε-valid layout satisfying some optimization criterion, it suffices to examine only the space of d.p. frameworks. Figure 4 shows a layout and its corresponding d.p. framework. Note that more than one d.p. framework can model a given layout.

While Lemma 1 correctly restricts the layout problem to the consideration of special spanning forests of $\mathcal{G}$, it is unfortunately not a tight characterization of all ε-valid layouts. That is, a d.p. framework over an overlap graph $\mathcal{G}$ built at error-rate ε does not necessarily lead to the construction, in the consensus phase of the computation, of a reconstruction R that confirms the layout to be *ε-valid*. Moreover, while the positional relationships between the fragment reads is nominally fixed by the d.p. framework, the precise intervals (s_i, e_i) of each read in the layout is not known until the third phase is completed. In essence, the difficulty is that a d.p. framework is only an approximation to the final result of the computation. In part this issue was addressed carefully in [Kec-91] where it is shown that the layout corresponding to a d.p. framework is no worse than $2\varepsilon/(1-\varepsilon)$-valid and that the lengths and positions implied by the d.p. framework (to be introduced) are off by no more than ε. These *worst-case* bounds assume that ε is small and that errors are uniformly distributed across overlaps. While not proven here we further contend that from a probabilistic point of view, the mean deviation between the the final positions and those of the d.p. framework is 0, and that the layouts modeled by d.p. frameworks are valid at mean rates very close to ε. Moreover, in practice the issue of the rate of validity is usually mute since ε is set to the *maximum* error rate, whereas the mean error rate in the overlaps of $\mathcal{G}$ are significantly smaller, e.g., a 10% maximum versus a 2% average. Hence for our current purposes, we will assume that the low-level "noise" in the fragment read data is an annoyance small enough that it does not affect the validity of the estimates on length, position, and error-rate of the layout and reconstruction resulting from a given d.p. framework[4].

With this assumption in hand we now turn to estimating the start positions of every fragment in the layout that will result from a given contig of a d.p. framework by considering the layout that would result if one assumed $\varepsilon = 0$. A formal description of the desired start-points is greatly simplified with some additional notation. Suppose $f \overset{\pi}{\rightleftharpoons} g$ is an edge in $\mathcal{G}$. Let $\pi.s_g$ denoted $\pi.s_A$ if $\pi.A = g$ and $\pi.s_B$ otherwise. Similarly define $\pi.e_g$, $\pi.s_f$, and $\pi.e_f$ so that we need not struggle with which fragment is playing the role of the A-fragment in the encoding of π. Next let $\pi.lft_g = min(\pi.s_g, \pi.e_g)$ and let $\pi.rgt_g = max(\pi.s_g, \pi.e_g)$ be the ordered delimiters of g's overlap substring, i.e., $g[\pi.lft_g, \pi.rgt_g]$. Finally, let $\pi.hang_g = |[1,|g|] - [\pi.lft_g, \pi.rgt_g]|$ be the number of symbols in g not in the overlap, and let $\pi.suf_g$ be true iff the overlap substring of g is a suffix of g, i.e., $\pi.rgt_g = |g|$. Note that for dovetail edges, $\pi.suf_g$ is true if and only if the arrowhead at g is directed away from g. Thus $\pi.suf_g$ effectively denotes the direction of the arrowhead of π at g.

Now suppose that a contig of a d.p. framework has as its dovetail path the sequence of reads and edges $f_1 \overset{\pi_1}{\rightleftharpoons} f_2 \overset{\pi_2}{\rightleftharpoons} f_3 \cdots \overset{\pi_{n-1}}{\rightleftharpoons} f_n$. Let sp_i denote our start point approximation for read f_i and let fwd_i be true iff f_i occurs in the forward (i.e.,

[4] Note however, that this "noise" is a crucial feature for the overlap and consensus phases of assembly where it cannot be treated cavalierly.

uncomplemented) orientation in the layout. Directly from the definitions of the edges π_i it follows that:

$$\begin{aligned} sp_1 &= 1 \\ fwd_1 &= \pi_1.suf_{f_1} \end{aligned}$$

and for $i > 1$:

$$\begin{aligned} sp_i &= sp_{i-1} + \pi_{i-1}.hang_{f_{i-1}} \\ fwd_i &= \neg\pi_{i-1}.suf_{f_i}. \end{aligned}$$

To place the contained reads of a d.p. framework, suppose that $g \overset{\pi}{\rightleftharpoons} f$ where f is contained in g whose orientation fwd_g and start point sp_g are known. Because g is the containing read it must be that $\pi.A = g$ by convention and it follows that:

$$sp_f = sp_g + \begin{cases} \pi.s_A & \text{if } fwd_g \\ |g| - \pi.e_A & \text{otherwise} \end{cases}$$

$$fwd_f = \begin{cases} fwd_g & \text{if } \pi.s_B \leq \pi.e_B \\ \neg fwd_g & \text{otherwise} \end{cases}$$

Let the set of start points and orientations so computed constitute the *stick layout* of the given d.p. framework.

Using the criterion of the previous section, the fragment assembly problem is equivalent to finding the d.p. framework whose stick layout minimizes the relative 2-sided Kolmogorov-Smirnov statistic δ. Figure 4 illustrates the start-point positions for a given d.p. framework. Suppose that $sp_1 \leq sp_2 \leq \cdots \leq sp_F$ is the ordered set of start points in a stick layout *including* those of contained fragment reads. Let the *uniform deviation* λ of a stick layout be:

$$\lambda = \frac{1}{2}\left(max_{1 \leq i \leq F}(\frac{i}{F} - \frac{sp_i}{G'}) - min_{1 \leq i \leq F}(\frac{i}{F} - \frac{sp_i}{G'}) + \frac{1}{F} \right)$$

Assuming that the source distribution is the uniform distribution, Lemma 2 shows that a stick layout of minimum uniform deviation maximizes likelihood in the sense of the 2-sided Kolmogorov-Smirnov statistic.

Lemma 2: If fragment reads are uniformly sampled from the target then a d.p. framework whose stick layout has minimal uniform deviation λ results in a reconstruction whose layout has minimum relative deviation δ.

Proof: All that need be shown is that the uniform deviation λ of a stick layout is indeed the relative deviation δ of the observed and uniform distributions discussed in the previous section on the 2-sided Kolmogorov-Smirnov statistic. First observe that

$$max_{1 \leq x \leq G'+1} |D_{obs}(x) - D_{uni}(x - \alpha)| =$$
$$max\left(max_{1 \leq i \leq F}(\frac{i}{F} - \frac{sp_i - \alpha}{G'}) \, , \, max_{1 \leq i \leq F}(\frac{sp_i - \alpha}{G'} - \frac{i-1}{F}) \right)$$

by the definitions of the distributions and because the extremes must occur at the "corners" of the *step-function* D_{obs}. But by algebra this is just:

$$max\left(max_{1 \leq i \leq F}(\frac{i}{F} - \frac{sp_i}{G'}) - \frac{\alpha}{G'} \, , \, -min_{1 \leq i \leq F}(\frac{i-1}{F} - \frac{sp_i}{G'}) + \frac{\alpha}{G'} \right)$$

Thus it follows that the optimum choice of the shift factor α is halfway between the *max* and the *min* giving the formula for the deviation of a stick layout presented above. □

We conclude this section with a description of the objective for d.p. frameworks that results in a shortest reconstruction. Let the *length* of an overlap π, $length(\pi)$, be $(|\pi.s_A - \pi.e_A| + |\pi.s_B - \pi.e_B|)/2$, the average length of the two overlapping substrings. Let the *weight* of a d.p. framework be the sum of the lengths of its edges. The following result was first proved independently by Tarhio & Ukkonen and Turner for the case where $\varepsilon = 0$. Thus for sufficiently small ε it is true that:

Lemma 3 [TaU-88, Tur-89]: A maximum-weight d.p. framework results in a reconstruction of minimum length.

Proof: First note that by the estimation of positions above, the length of the reconstruction for a given d.p. framework is approximately $sp_n + |f_n|$ where f_n is the last fragment read in the dovetail path of the framework. But by unwinding the recursive definition of sp_n this is simply:

$$\Sigma_{i=1}^{n-1} \pi_i.hang_{f_i} \; + \; |f_n|$$

Finally, using the facts that (a) $\pi_i.hang_{f_i} \approx |f_i| - length(\pi_i)$, and (b) $length(\pi) \approx |g|$ when π is a containment edge and g is the contained fragment, it follows that the above equals:

$$\Sigma_f |f| \; - \; \Sigma_\pi length(\pi)$$

But the second summation is the weight of the d.p. framework and the first is constant. Thus maximizing weight minimizes length. □

4 Simplifying the Layout Problem

While we have described criteria for determining layouts that are shortest or have minimum relative δ, we have yet to describe effective algorithms for finding the appropriate d.p. frameworks. Typical overlap graphs involve up to 1000 fragment reads and so represent formidable instances of the NP-complete combinatorial problems involved [Kec-91, KeM-95]. We proceed here to detail a series of reductions to the overlap graph that reduce the number of edges and vertices without changing the space of potential solutions. The intuition is to assemble all portions of the problem that join in only one unique way, and then deal with the clearly delineated combinatorial choices that remain in the reduced *chunk graph*.

Consider then the following three transformations illustrated in Figure 5. The transformations are applied independently and in the order given:

1. *Contained Read Removal*: Every fragment read that is contained by another is removed from $\mathcal{G}$ along with all edges incident to it. A list of these reads is stored so that they may be reintroduced later into a tentative layout. Call the reduced graph $\mathcal{G}_1$ and note that all remaining edges are dovetail edges.

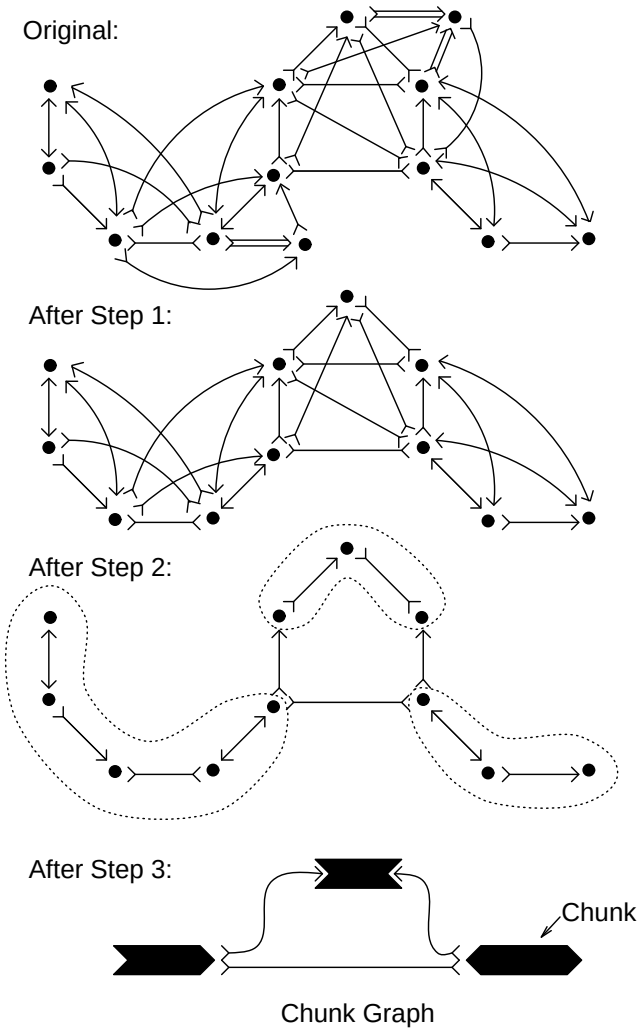

Fig. 5. Collapsing an Overlap Graph.

2. *Transitive Edge Removal*: If $f \overset{\tau}{\rightleftharpoons} g \overset{\tau'}{\rightleftharpoons} h$ and $f \overset{\pi}{\rightleftharpoons} h$ are *mutually consistent* overlaps among reads f, g, and h, then the edge for π is removed. Informally overlaps are mutually consistent if the overlap between f and h implied by the concatenation of the alignments τ and τ' is the same as that of π to within error rate ε, i.e. $\pi \approx_\varepsilon \tau \circ \tau'$. A simple, position-based formulation that we have found sufficient for practical purposes is:

$$\begin{aligned}
\tau.suf_g &\neq \tau'.suf_g \\
\pi.suf_f &= \tau.suf_f \\
\pi.suf_h &= \tau'.suf_h \\
\tau.hang_f + \tau'.hang_g &\in \pi.hang_f \pm (\varepsilon \cdot length(\pi) + \alpha) \\
\tau.hang_g + \tau'.hang_h &\in \pi.hang_h \pm (\varepsilon \cdot length(\pi) + \alpha)
\end{aligned}$$

where α is a small constant that helps capture fluctuations in the distribution of errors when the overlap length of π is small. Choosing α to be 3 is generous in practice. If desired, a more stringent definition of consistency could be developed around the actual alignments encoded in $\pi.\Delta$, $\tau.\Delta$, and $\tau'.\Delta$[5]. Call the resulting graph $\mathcal{G}_2$.

3. *Unique-Join Collapsing*: If there is an edge $f \overset{\pi}{\rightleftharpoons} g$ such that for every other edge $x \overset{\tau}{\rightleftharpoons} f$ adjacent to f, $\pi.suf_f \neq \tau.suf_f$, and for every other edge $x \overset{\tau}{\rightleftharpoons} g$ adjacent to g, $\pi.suf_g \neq \tau.suf_g$, then collapse f and g into a single vertex representing the *contig* of f overlapped to g by π. More informally the conditions for collapsing are that the arrowheads at f of all edges adjacent to f except π must point in the opposite direction to that of π, and the same must be true of arrowheads at g. Call the graph resulting from performing all such collapsings the *chunk graph* $\mathcal{G}_3$ and call the uniquely joined fragment reads represented by each vertex a *chunk*.

If the objective is to produce a shortest reconstruction then Step 1 above is conservative because the best way to utilize a contained fragment read f is within any of its containers as these must be substrings of any reconstruction. However, when the goal is the minimum relative δ reconstruction then contained reads are important as the issue is not just string content but also start-point spacing including those of the contained reads. Thus it is important where such contained reads are placed in a layout in the event there is more than one choice. Even more subtle is the very rare situation where a read f is a contained fragment read, say in read p, but where f is used in an orthogonal dovetail fashion in a solution, i.e., f joins two non-overlapping reads g and h (dovetail usage), and p does not overlap g or h (orthogonality to containment). For example, $g = xxxxxaaa$, $f = aaabbb$, $h = bbbyyyyy$, and $p = tttaaabbbttt$. Given the typical coverage of assembly problems, we very rarely encounter this situation, and usually have only a small amount of latitude in where to place contained reads. Thus we remove them in Step 1, and then optimally reincorporate them into the layouts proposed over the chunk graph $\mathcal{G}_3$.

In Step 2 one must be careful to first mark all transitive edges and then remove all marked edges. This is because one transitive edge may imply another is transitive, and so an arbitrary sequential strategy will fail as such a process is not Church-Rosser [ChR-36]. The critical feature of this transformation is that if $f \overset{\pi}{\rightleftharpoons} g$ is removed then there is still a dovetail path in $\mathcal{G}_2$ that places f and g in the same relative position and so produces the same portion of the reconstructed string (modulo the error rate ε) that would be produced by overlapping f and g directly. Thus, like Step 1, *this reduction is conservative with respect to string content but not necessarily start-point spacing*. Note next that if one were to separate such a tighly overlapping set of fragment reads on the path from f to g in order to obtain a better start-point spacing, one creates a reconstruction in which the string modeled by the overlapping

[5] For example, compute the alignment that is the concatenation of τ and τ', and then insist that the "distance" between the path of this alignment and that of π in an edit graph between f and h are within some limit.

part of $f \overset{\pi}{\rightleftharpoons} g$ is *repeated* in the reconstruction. Conversely, when the target has a repeated substring, fragment reads within the repeat form such tightly-overlapping paths as reads from different copies of the repeat cannot distinguish which copy they came from on the basis of ε-overlaps alone. As will be seen, this requires for the case of minimum relative δ reconstructions that *non-simple* paths be considered and that repeated portions of such paths be segregated to produce the best possible start-point spacing *after the fact*.

Unlike Step 2, the collapsing of Step 3 is Church-Rosser and so may take place sequentially in any order. Note that while initially one joins two reads together to form a chunk, one eventually begins to join chunks together to form yet larger chunks. Thus each chunk represents a dovetail path of fragment reads, and as such has two end reads. For a chunk $C \equiv f_1 \overset{\pi_1}{\rightleftharpoons} f_2 \cdots f_{t-1} \overset{\pi_{t-1}}{\rightleftharpoons} f_t$, let $C.lf = f_1$ be its left-end read and $C.rf = f_t$ be its right-end read. By the nature of the collapsing rule, the only edges in $\mathcal{G}_3$ adjacent to C are those adjacent to the suffix or prefix of $C.lf$ and $C.rf$ extending to the left and right, respectively. More formally let $C.le = \{f \overset{\tau}{\rightleftharpoons} g : f = C.lf\}$ be the set of all edges adjacent to the left end of C (not including the one edge, π_1, which is internal to C). Similarly define $C.re$, the right-end adjacency set. It follows by the construction that $\tau.suf_{C.lf} = \neg\pi_1.suf_{C.lf}$ for every edge $\tau \in C.le$, and also that $\tau.suf_{C.rf} = \neg\pi_{t-1}.suf_{C.rf}$ for every edge $\tau \in C.re$. That is, the edges adjacent to an end all have the same arrowhead direction that is the opposite of that of the edge adjacent to the end read but internal to the chunk.

These last observations are reflected in the chunk graph of Figure 5 as follows. First, chunk vertices are not depicted as the usual circle of a conventional graph, but are drawn as solid bars with an angled notch or angled protrusion depending on the orientation of the chunk's end reads. Second, edges emanate from one end of a chunk-bar or the other and their arrowhead directions always complement the notch of the end they emanate from as they must by construction. To further reenforce these facts, one could imagine starting Step 3 by converting $\mathcal{G}_2$ into a chunk graph as follows: (1) map every fragment-read vertex to a chunk vertex that is a bar with notch at one end and a protrusion at the other, and (2) reattach the end of an edge adjacent to a vertex to the end of the vertices' bar that complements the edge's arrowhead. That is, let each vertex f be a chunk C_f where $C_f.lf = C_f.rf = f$ and arbitrarily divide its adjacent edges into $C.le = \{f \overset{\pi}{\rightleftharpoons} g : f = C.lf \text{ and } \pi.suf_f\}$ and $C.re = \{f \overset{\pi}{\rightleftharpoons} g : f = C.lf \text{ and } \neg\pi.suf_f\}$. Given this initial chunk graph, the collapsing rule of Step 3 is simply: collapse $C_1 \overset{\pi}{\rightleftharpoons} C_2$ if and only if $C_1.re = C_2.le = \{\pi\}$. In the event the chunks are collapsed to form C', then $C'.lf = C_1.lf$, $C'.le = C_1.le$, $C'.rf = C_2.rf$, and $C'.re = C_2.re$. Finally, a *chunk path* in a chunk graph is any sequence $C_0 \overset{\pi_1}{\rightleftharpoons} C_1 \cdots C_{n-1} \overset{\pi_{n-1}}{\rightleftharpoons} C_n$ of chunk vertices and edges such that $\pi_i \in C_{i-1}.re \cap C_i.le$. More informally, a chunk path is a walk that enters a chunk bar at one end and exits the other. Note that unlike the overlap graph $\mathcal{G}$, one need not be concerned with arrowhead directions in the chunk graph $\mathcal{G}_3$ because its construction guarantees proper joining. Indeed, a chunk path in $\mathcal{G}_3$ corresponds to a dovetail path in $\mathcal{G}$ when one expands each chunk vertex back into the dovetail subpath it models.

The original impetus for the design of the collapsing process just described was interactions with Ron Lundstrom, Jim Orlin, and Hershel Safer in the Summer of

'93. Both groups have since been exploring this basic idea. These colleagues have been working with a more conservative collapsing strategy that replaces Steps 2 and 3 with a single step that joins fragment reads f and g along overlap π if and only if (a) $\{h : h \overset{\tau}{\rightleftharpoons} f \text{ and } \tau.suf_f \neq \pi.suf_f\} \subseteq \{h : h \overset{\tau}{\rightleftharpoons} g \text{ and } \tau.suf_g = \pi.suf_g\}$, and (b) $\{h : h \overset{\tau}{\rightleftharpoons} g \text{ and } \tau.suf_g \neq \pi.suf_g\} \subseteq \{h : h \overset{\tau}{\rightleftharpoons} f \text{ and } \tau.suf_f = \pi.suf_f\}$ [LOS-94]. This illustrates that there is more than one possible way to exploit the basic idea and it will be interesting to compare various developments of this kind. This author chose the most aggressive collapsing strategy possible, believing that reduction of problem size outweighs the complexity of the expression of the objective in the reduced problem.

We now turn to characterizing shortest and minimum relative δ reconstructions in terms of paths in a chunk graph. The collapsing rules of Steps 1 to 3 are all conservative with regard to string content but not to the modeling of fragment-read positions in a layout. The latter is obvious because (1) reads were removed in Step 1, and (2) some joining possibilities were removed in Step 2. It is thus easiest to start by considering the characterization of a shortest reconstruction string. To do so it is necessary to make sure that a few edges that may have been removed in Step 2 are in the penultimate graph. An edge $f \overset{\pi}{\rightleftharpoons} g$ in $\mathcal{G}_1$ is *end-to-end* if in the chunk graph $\mathcal{G}_3$ f and g are in $\bigcup_{C \in \mathcal{G}_3} C.lf \cup C.rf$, i.e., if f and g are the end reads of chunk vertices. Add all the end-to-end edges back into $\mathcal{G}_3$ (if necessary) and call the resulting graph $\mathcal{G}_4$. Note that in practice it is quite unlikely a transitively removed edge is an end-to-end edge and so very few edges need to be added back to form $\mathcal{G}_4$. However, in theory they are critical to the following analog to Lemma 3 which shows that in order to find the shortest superstring of all the reads, the objective in $\mathcal{G}_4$ is the same as in the original uncollapsed overlap graph.

Lemma 4: A simple chunk path of maximal weight in $\mathcal{G}_4$ results in a shortest reconstruction.

Proof: Consider an ε-valid layout corresponding to a shortest reconstruction string. The string is known (up to accuracy ε) on the basis of just the dovetail path(s) of the layout's corresponding d.p. framework. Suppose the sequence of fragment reads on this dovetail path is $f_0, f_1, \cdots f_n$. We say the shortest reconstruction is modeled by a dovetail path in $\mathcal{G}$.

Suppose one of the reads, say f_k, in the path above is removed by Step 1. By induction f_k must be contained in some fragment read not removed in Step 1 and consequently in the path, say it is f_j. If f_{k-1} does not overlap f_{k+1} (or one of them doesn't exist), then a shorter string would be obtained by breaking the path into two contigs $f_0 f_1 \cdots f_{k-1}$ and $f_{k+1} f_{k+2} \cdots f_n$, and folding f_k into the portion of the string containing f_j, a contraction. Thus f_{k-1} and f_{k+1} must overlap. But then we may remove f_k from the chain without changing the string being modeled by letting f_{k-1} and f_{k+1} be linked directly with the edge between them. Thus we can conclude that a shortest string is modeled by a dovetail path in $\mathcal{G}_1$.

Now suppose that one of the edges $f_{k-1} \overset{\pi}{\rightleftharpoons} f_k$ in the dovetail path of $\mathcal{G}_1$ is removed by Step 2. As noted earlier, there must still be some path in $\mathcal{G}_2$ from f_{k-1} to f_k that models the same string as that of the join of the two fragment reads.

To confirm this path property it suffices to note that (1) the property remains true when a single transitive edge is removed, and (2) there is an order of such single step removals leading to $\mathcal{G}_2$ because the reduction is non-cyclic (i.e. a transitive edge cannot cause the removal of an edge which implies its removal). So consider replacing every edge between two reads in the chain that is not in $\mathcal{G}_2$ by the path in that graph that transitively implies it. Note carefully that the resulting path may not necessarily be simple. Nonetheless, we can conclude that a shortest string is modeled by a (not necessarily simple) dovetail path in $\mathcal{G}_2$.

Next it is easy to see that a dovetail path in $\mathcal{G}_2$ maps directly to a chunk path in $\mathcal{G}_3$, as the converse would violate the contraction of Step 3. For example, if a read f_k were in a chunk and had g as its predecessor in the chunk, then $g \neq f_{k-1}$ would imply f_k had two predecessors and so would not have been joined to g, a contradiction. Thus a shortest string is modeled by a (not necessarily simple) chunk path in $\mathcal{G}_3$.

Finally, suppose the chunk path in $\mathcal{G}_3$ is not simple. It must then have a maximal repeated subpath $C_jC_{j+1}\cdots C_{j+r} = C_kC_{k+1}\cdots C_{k+r}$ where $j < k$, $C_{j-1} \neq C_{k-1}$, and $C_{j+r+1} \neq C_{k+r+1}$. If C_{k-1} does not overlap C_{k+r+1} then a shorter string is obviously obtained by breaking the path into two contigs $C_0C_1\cdots C_{k-1}$ and $C_{k+r+1}C_{k+2}\cdots C_n$, a contradiction. On the other hand if they do overlap then the edge modeling the overlap is in $\mathcal{G}_4$ as every end-to-end edge is in this augmentation of the chunk graph $\mathcal{G}_3$. Thus the repeat may be excised and C_{k-1} joined directly to C_{k+r+1} without changing the string being modeled. Thus a shortest string is modeled by a simple, proper chunk path in $\mathcal{G}_4$.

So it has been shown that to find a shortest reconstruction one may restrict attention to the simple, chunk paths of $\mathcal{G}_4$. It then follows that because chunks are formed along unique joins, the only way to minimize length is to maximize the overlap of the edges used to connect the chunks. Ergo the statement of the lemma. □

The proof of Lemma 4 provides the insights needed to characterize layouts (as opposed to strings) in chunk graph terms. Let $\mathcal{F}$ denote the set of fragment reads and let $\mathcal{C} \subseteq \mathcal{F}$ be the set of contained reads removed in Step 1. The chunk graph is a representation of the reads in $\mathcal{F} - \mathcal{C}$ and the overlaps between them. A *chunk framework over* $\mathcal{F} - \mathcal{C}$ is (1) a collection of chunk paths in the chunk graph $\mathcal{G}_3$, and (2) for each chunk, say C, that is repeated $k > 1$ times in the chunk path collection, a partition of C's reads into k classes and a bijective assignment of the classes to the k copies of C in the chunk paths. To capture the placement of contained reads $\mathcal{C}$ in a layout it is necessary to revert to the the original overlap graph $\mathcal{G}$. A *chunk framework over* $\mathcal{F}$ is a chunk framework over $\mathcal{F}-\mathcal{C}$ in the graph $\mathcal{G}_3$ with the addition of (3) dovetail paths in $\mathcal{G}$ connecting the free ends of two chunk paths, all of whose internal reads are in $\mathcal{C}$, and (4) a containment forest in $\mathcal{G}$ whose non-root reads are in $\mathcal{C}$ less those used in part (3).

Lemma 5: Every ε-valid layout is modeled by a chunk framework over $\mathcal{F}$.

Proof: By Lemma 1 every ε-valid layout is modeled by a d.p. framework over $\mathcal{F}$. Remove from the d.p. framework all reads in $\mathcal{C}$. Clearly the containment forest

part of this framework is removed and constitutes part (4) of the chunk framework. Also, but much more rarely, a read or two in $\mathcal{C}$ may be removed from the dovetail path portion of the d.p. framework. These reads constitute part (3) of the chunk framework. What remains is a dovetail path(s) of reads in $\mathcal{F} - \mathcal{C}$. Take this dovetail path and map it through the transformations of Lemma 4 to the (non-simple) chunk path(s) in the chunk graph $\mathcal{G}_3$ that models the same *string*. The layout corresponding to this chunk path may not be the one modeled by the dovetail path as it can have *additional* reads in $\mathcal{F} - \mathcal{C}$ overlapped between the reads of the dovetail path in order to model the transitive edge between them that was removed in Step 2. Note these additional reads must be repeated instances as every read occurred once in the original dovetail path. In fact, suppose chunk $C \equiv f_0 \overset{\pi_1}{\rightleftharpoons} f_1 \overset{\pi_2}{\rightleftharpoons} f_2 \cdots f_{t-1} \overset{\pi_t}{\rightleftharpoons} f_t$ occurs in the chunk path k times and label the copies $C^{(1)}, C^{(2)}, \cdots C^{(k)}$. Each instance $C^{(i)}$ models a portion

$$\mathcal{P}(C^{(i)}) = f_{j_0^{(i)}} \overset{\tau_1^{(i)}}{\rightleftharpoons} f_{j_1^{(i)}} \cdots f_{j_{t_i-1}^{(i)}} \overset{\tau_{t_i}^{(i)}}{\rightleftharpoons} f_{j_{t_i}^{(i)}}$$

of the original dovetail path, where $f_{j_0^{(i)}}, f_{j_1^{(i)}}, \cdots f_{j_{t_i}^{(i)}}$ is a subsequence of $f_0, f_1, \cdots f_t$ because the transitive edge $\tau_g^{(i)}$ is now modeled by the sequence of edges $\pi_{j_{g-1}+1}^{(i)} \circ \pi_{j_{g-1}+2}^{(i)} \circ \cdots \circ \pi_{j_g}^{(i)}$. Observing that $\{\mathcal{P}(C^{(i)})\}_i$ is a *k-partition* of the reads in C, the result follows. $\square$

To this point we have proceeded formally. On casual inspection it may appear that we have replaced the simple formulation of Lemma 1 with the more complex formulation of Lemma 5. But the real power of what has been done becomes clear in the arena of practice. First and most important is the fact that in practice the chunk graph is a very small graph. For example, in 20 trials on target sequences of length 40,000 that are the result of coin tosses and where sampled reads have a 5% error rate introduced into them, we found that in all cases but one the chunk graph had no edges! That is, the solution consisted of contigs corresponding to each isolated chunk vertex in the graph. In the one case where the collapsing was not complete there were three edges and the solution was immediately clear by inspection. Thus the chunk graph construction provides a very robust and simple algorithm for sequencing projects involving non-repetitive DNA. On target sequences that have repeats, one again sees almost complete collapsing *except* around the repeated blocks. For example Figure 6 shows a stylized example of what happens with our running example of Figures 1 and 2. In practice we move from a graph with on the order of a thousand vertices to one with generally less than fifty, often less than ten, depending on the number and size of repeated substrings in the target sequence from which reads are sampled. The cost of a somewhat more complex solution description is far outweighed by the resulting combinatorial reduction.

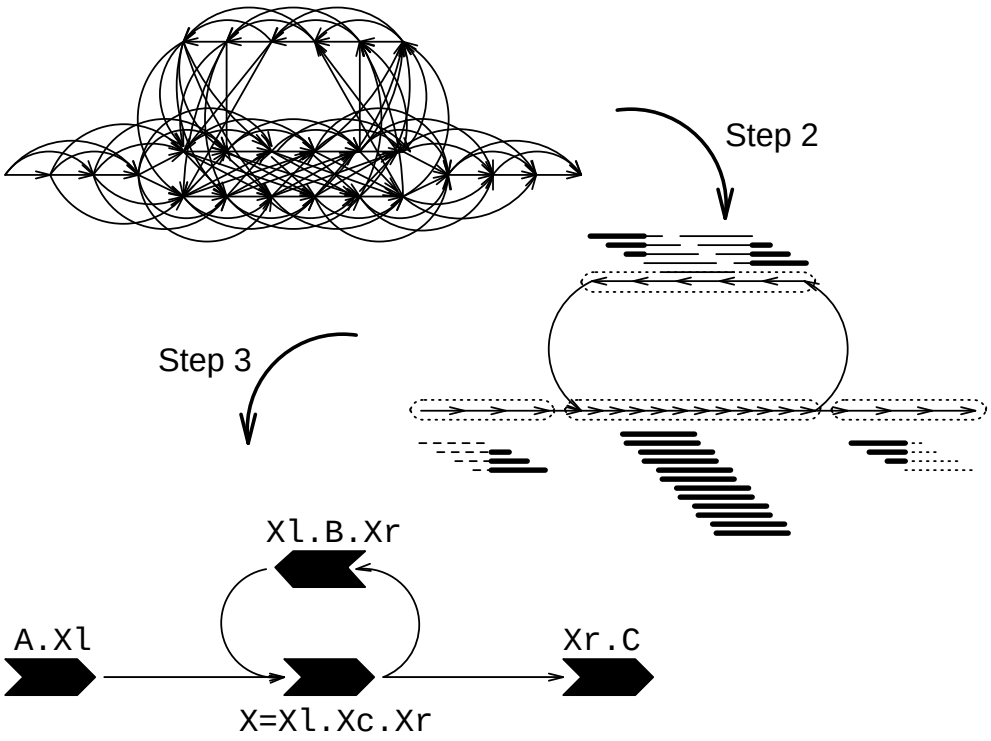

Fig. 6. Schematic Collapsing of a Target with a Repeat.

Several further practical observations help us even more. First failures to collapse are indicative of repeats. Also recall from earlier discussion that repeats collapse into the same chunk and are then apparent because the distribution of start points within the chunk is too high. That is, the chunk, considered as a layout in isolation, gives higher Kolmogorov-Smirnov likelihood values if one assumes the target is of length G'/k for some $k > 1$. Indeed the values of k giving high likelihood values delimit the set of possible multiplicities of the repeat and thus the number of times its chunk may be repeated in a chunk path. Thus we may limit our solution search to chunk paths in which each chunk may be traversed a number of times dependent on a liberal reading of the likelihood of its being a repeat. Moreover most chunks are expected to appear once on a path. Next, note that given a chunk path, the partition of any repeated chunk is in practice always clear from the correlated-differences among overlapping reads. This is because DNA repeats are never perfect in nature but usually vary at least 5% from copy to copy. Thus even while there may be 5% noise in the fragment read data due to errors, one can still discern different copies (and also get further confirmation as to their number) by finding columns of the implied multi-alignment where the overlapping reads clearly partition themselves consistently over such columns. Thus finding the partitioning of a repeated chunk is quite evident or limited to only a few possibilities.

We are currently developing a branch-and-bound procedure that searches the

space of chunk frameworks looking for layouts that are maximally-likely in the sense of the Kolmogorov-Smirnov statistic. The details of this procedure are not sufficiently resolved for this paper, but its essential features are as follows. First, a repetition factor or range of factors is assigned to each chunk according to layout density and partitioning possibilities. Then, a branch-and-bound procedure searches the space of *all* chunk paths consistent with the factor ranges. For each chunk framework over $\mathcal{F} - \mathcal{C}$ reached by the B&B procedure, the algorithm goes on to explore the possible completions with fragment reads from $\mathcal{C}$. First these reads are examined for joining chunk paths (usually no such joins are possible), and then the remaining reads are incorporated in a containment fashion, where again possibilities are limited to just a few by the removal of transitive containment edges in analogy to what has been done for dovetail edges in Steps 2 and 3. The resulting solutions are evaluated with respect to their relative deviations δ and the best ones stored for reporting at the end of the search.

As final note, we observe that it is easy to incrementally compute the uniform deviation $\lambda = \delta$ of a layout as its pieces are being joined together. For a given chunk C containing reads $\{f_k\}$ and whose layout places these reads at start positions $\{sp_k\}$, let $C.span = sp_{|C|} + |f_{|C|}|$, $C.\max = \max_k\{\frac{k}{F} - \frac{sp_k}{G'}\}$, and $C.\min = \min_k\{\frac{k}{F} - \frac{sp_k}{G'}\}$. Note that in these terms the uniform deviation of the layout for chunk C is $\frac{1}{2}(C.\max - C.\min + \frac{1}{F})$. Now suppose that $C_1 \overset{\pi}{\rightleftharpoons} C_2$ and the two chunks are to be merged into a single chunk C' along this overlap. It is an easy exercise to see that $C'.span = C_1.span + C_2.span - length(\pi)$, $C'.\max = \max(C_1.\max, C_2.\max + \Delta)$, and $C'.\min = \min(C_1.\min, C_2.\min + \Delta)$ where $\Delta = \frac{|C_1|}{F} - \frac{C_1.span - length(\pi)}{G'}$. In this way the uniform deviation of a current set of choices leading to a chunk path may be maintained in constant time overhead and can direct the B&B search. Further note that this observation may also be used to compute the deviations of the chunks as they are built in Step 3 of the collapsing sequence. Finally, to accommodate chunks that are repeated $r > 1$ times, it suffices to simply extend their internal span to $C.span = r \cdot sp_{|C|} + |f_{|C|}|$ as if the repeated chunk had been unbraided, and to correspondingly set $C.\max = \max_k\{\frac{k}{F} - \frac{r \cdot sp_k}{G'}\}$, and $C.\min = \min_k\{\frac{k}{F} - \frac{r \cdot sp_k}{G'}\}$.

Acknowledgement

I would like to thank Ron Lundstrom and Hershel Safer at Genome Therapeutics, and Jim Orlin at the Sloan School of Management, MIT, for the stimulating interactions that were pivotal to this work. I also wish to thank my associate Susan Larson for preliminary implementations of the ideas, and to my colleague Pete Downey for introducing me to the work of Kolmogorov and Smirnov.

References

[Bel-92] Bell, G.I. 1992. Roles of repetitive sequences. *Computers Chem.* 16, 135-143.

[Bir-52] Birnbaum, Z.W. 1952. Numerical tabulation of the distribution of Kolmogorov's statistic for finite sample size. *J. Amer. Statist. Assoc.* 47, 425-441.

[ChR-36] Church, A. and Rosser, J.B. 1936. Some properties of Conversion. *Trans. Amer. Math. Soc.* 39, 472-482.

[Hua-92] Huang, X. 1992. A contig assembly program based on sensitive detection of fragment overlaps. *Genomics* 14, 18-25.

[Iri-94] 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.

[Kec-91] Kececioglu, J. 1991. Exact and Approximation Algorithms for DNA Sequence Reconstruction. Ph.D. thesis, Technical Report 91-26, Department of Computer Science, The University of Arizona, Tucson, Arizona 85721.

[KeM-95] Kececioglu, J. and Myers, E. 1995. Exact and Approximate Algorithms for the Sequence Reconstruction Problem. *Algorithmica* 13, to appear.

[Kol-33] Kolmogorov, A. 1933. Sulla determinazione empirica di una legge di distribuzione. *Giorn. Ist. Ital. Attuari.* 4, 83-91.

[LMS-95] Landau, G.M., Myers, E.W. and Schmidt, J.P. 1995. Incremental String Comparison. *SIAM J. on Computing*, submitted.

[LaW-88] Lander, E.S. and Waterman, M.S. 1988. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* 2, 231-239.

[Li-90] Li, M. 1990. Towards a DNA sequencing theory *Proc. 31st IEEE Symp. on Found. of Computer Science*, 125-134.

[LOS-94] Lundstrom, R., Orlin, J., and Safer, H. 1994. *Personal communication.*

[MaG-77] Maxam, A.M. and Gilbert, W. 1977. A new method for sequencing DNA. *Proc. Natl. Acad. Sci. USA* 74, 560-564.

[Mye-86] Myers, E. 1986. Incremental Alignment Algorithms and Their Applications. Tech. Rep. 86-22, Dept. of Computer Science, U. of Arizona, Tucson, AZ 85721.

[PSU-84] Peltola, H., Söderlund, H. and Ukkonen E. 1984. SEQAID: a DNA sequence assembly program based on a mathematical model. *Nucleic Acids Research* 12, 307-321.

[RoH-94] Rowen, L., and L. Hood, L. 1994. *Personal communication.*

[Sel-80] Sellers, P.H. 1980. The Theory and Computation of Evolutionary Distances: Pattern Recognition. *J. Algorithms* 1, 359-373.

[Smi-41] Smirnov, N.V. 1941. Approximate laws of distribution of random variables from empirical data. *Uspekhi Mat. Nauk* 10, 179-206. (In Russian.)

[TaU-88] Tarhio, J. and Ukkonen, E. 1988. A greedy approximation algorithm for constructing shortest common superstrings. *Theoretical Computer Science* 57, 131-145.

[Tur-89] Turner, J. 1989. Approximation algorithms for the shortest common superstring problem. *Information and Computation* 83, 1-20.

[SNC-77] Sanger, F., Nicklen, S. and Coulson, A.R. 1977. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA* 74, 5463-5467.

[SCH-82] Sanger, F., Coulson, A.R., Hong, G.F., Hill, D.F. and Petersen, G.B. 1982. Nucleotide sequence of bacteriophage λ DNA. *J. Mol. Biol.* 162, 729-773.

This article was processed using the LaTeX macro package with LLNCS style