Integration of Competing Ancillary Assertions in Genome Assembly

Christian Burks
Theoretical Biology
and Biophysics Group
T-10, MS K710
Los Alamos National Laboratory
Los Alamos, NM 87545
cb@t10.lanl.gov

Rebecca J. Parsons
Computer Research
Applications Group
C-3, MS B265
Los Alamos National Laboratory
Los Alamos, NM 87545
rebecca@lanl.gov

Michael L. Engle
Theoretical Biology
and Biophysics Group
T-10, MS K710
Los Alamos National Laboratory
Los Alamos, NM 87545
mle@t10.lanl.gov

Abstract

Assembly of genomic sequences and maps relies on a primary set of experimental data (e.g., the sequences of individual DNA fragments, or hybridization fingerprints of individual clone inserts), but almost always also relies on several streams of related but distinct kinds of data for completeness and accuracy of the final construction. These secondary data sets, which we term ancillary information, usually contain errors (as do the primary data sets, therefore creating the possibility of conflict between data sets), often arise from different experimental protocols and correspond to different scales of measurement, and occasionally include non-quantitative statements about the data. We present an approach for integration of ancillary assertions in the optimization of genome assembly, based on simultaneous balancing among the primary and secondary data sets, and include specific examples in the context of assembling DNA sequencing fragments to reconstruct a parent sequence.

Genome Assembly

Mapping the human and other genomes involves many strategies and scales. However, most of these approaches have in common the methodology of individually characterizing small pieces of the object being mapped, evaluating the pairwise similarity of the pieces' characteristic(s), and assembling the pieces into an interleaved tiling, or *layout*, from which can be derived an overview map of the relations of the pieces to each other (and to the original, parent object). Exploration of all possible assemblies, given the antecedent data, is combinatorially complex and therefore computationally prohibitive for data sets involving large numbers of pieces. Both the evaluation of pairwise relationships and the optimization of assembly are further complicated by the typical use of multiple kinds and sources of input data and the universal presence of error in the input data sets.

We focus here on the context of large-scale DNA sequencing (Hunkapiller *et al.* 1991; Venter 1994), the highest-resolution approach to genome mapping, and in particular on integrating the effects of different kinds of ancillary information from multiple sources into the optimization of genome assembly by simultaneously balancing among the primary and secondary data sets used to influence the layout of fragments. A DNA *sequence* is represented by a string of characters drawn from a four-letter alphabet (A, C, G, and T) corresponding to the four *bases* found in the DNA polymer. A piece, or *fragment*, corresponds in our context to a string of 100-1000 bases; overlap strength and offset between pairs of fragments is based on character string comparison. The output overview map represents a consensus sequence on the order of 1000-1,000,000 bases, generated by voting in aligned columns of bases resulting from the assembly of the input fragments.

Large-scale sequencing

A number of groups have recently published the results of large-scale sequencing projects generating from tens of thousands to millions of contiguous bases (e.g., (Wilson *et al.* 1994)). These recent efforts are noteworthy because of their conception and implementation as short-term, globally-comprehensive sequencing projects; this contrasts with previously published sequences of comparable size, which have been the result of piecing together many smaller projects. The limitation of experimental methods to stretches of 100-1000 contiguous bases for direct determination of DNA sequences has meant that longer regions of DNA have to be determined as shorter, overlapping fragments. For a large project, involving hundreds or thousands of sequence fragments, the computation of an optimal layout requires alternatives to the systematic exploration of all possible assemblies. The most prevalent algorithmic approach is the greedy construction of a single or a few solutions (e.g., (Dear & Staden 1991; Huang 1992)), though others have experimented with stochastic search (e.g., (Burks *et al.* 1994; Churchill *et al.* 1993; Parsons, Forrest, & Burks 1993; 1994)) or rapid approximations to exact constructions (e.g., (Kececioglu & Myers 1989; Kececioglu 1991)). These algorithms have recently been reviewed by My-

ers (Myers 1994).

A number of other factors further complicate the computational complexity of sequence assembly. The input fragment sequences usually include experimental ambiguities or errors that affect assessment of pairwise overlap strength and subsequent detailed alignments. Though sequencing both strands contributes to minimizing errors arising in base calling, it also leads to the necessity for assigning and tracking strand sense through the assembly and alignment calculations. Naturally-occurring DNA sequences tend to be repetitive on many different scales. Repetitive sequences longer than individual fragments may cause ambiguities in the sequence assembly that cannot be resolved without additional information; this problem is exacerbated by higher rates of conservation among and larger repeat units in a repeat family. Finally, the rates and distributions of errors can vary with source (e.g., different protocols, technicians, and sources of material).

Optimization with ancillary assertions

Several potential sources of ancillary information are available to compensate for either incompleteness or errors in the primary sequence data. For example, subsets of a fragment set may be known to be proximal to one another by virtue of previous, independent characterization of sub-cloned regions of the insert being sequenced. At the same time, it is desirable to avoid implementing these ancillary data as absolute constraints because they too will often contain errors. Ideally, one would like to integrate their impact into the overall objective function driving the optimization of layout.

The typical approach to sequence assembly involves reliance on a well-defined algorithm to automatically generate an optimal layout based only on the underlying, primary sequence data, usually followed by a lengthy manual editing process to incorporate the ancillary information that the experimentalist has at hand. We present here a new strategy based on the integration of these ancillary assertions into the initial, automated layout, and present three examples of how such ancillary data can be built into and influence the objective functions controlling the assembly optimization. In developing this strategy, we plan to develop a limited library of possible functions representing characteristics of, or relations between, the pieces being assembled so that new sources and kinds of ancillary information can be implemented by casting the new kinds of information in terms of an existing library of functions.

Systems, Data and Software

The software was developed on a Sun Microsystems (Mountain View, CA) workstation running SunOS UNIX. Programs were written in the C programming language, and the interface was implemented on the OpenWindows (X-11 compatible) platform.

All artificial fragment sets were generated by extracting known nucleotide sequences from GenBank (Burks *et al.* 1992) and fragmenting them computationally, using **genfrag** (v. 2.0), to conform to a range of desired values for coverage, fragment length, repeat density, and error rate (Engle & Burks 1993; 1994).

Sequence fragments were assembled into output consensus sequences using the following modules:

score was used for scoring pairwise overlap strengths; it counts the number of identical words along each diagonal in the string comparison matrix, and has been described previously (Churchill *et al.* 1993).

layout takes a given ordered list of the fragments, permutes them, and evaluates a corresponding objective function to optimize layout; though we have experimented with several approaches, we typically use self-adaptive annealing to drive the optimization (Burks *et al.* 1994). The permutations are based on alternating sets of randomly-selected sequential pairwise exchanges of ordinal assignments in the list (Churchill *et al.* 1993) with sets of randomly-selected transpositions and inversions of neighboring blocks of ordinal assignments in the list (Burks *et al.* 1993). For straightforward layout determination without use of ancillary information, this optimization is driven by an objective function, F_{seq},

$$F_{seq} = \sum_{i=1}^{N} \sum_{j=1}^{N} |p_i - p_j| s_{i,j} \qquad (1)$$

where p_i and p_j are the ordinal assignments of fragments i and j, and $s_{i,j}$ is their overlap strength (Churchill *et al.* 1993). The addition of ancillary assertions is mediated by adding other objective function components to this component.

Finally, **mfa** generates a multiple alignment by performing a series of global alignments using dynamic programming on overlapping successive pairs of overlapping fragments in the layout (Engle, Parsons, & Burks 1994), using the order of neighboring fragments generated by **layout**. The final step of generating a consensus from the multiple sequence alignment is accomplished using a simple column majority to call a base at each position. The quality of this final consensus sequences is assessed by examining percent match to and percent coverage of the initial parent sequence using the **align** program (Pearson 1993).

General Approach and Sample Implementations

Sequencing efforts frequently require significant manual editing that relies on information other than the overlap strengths used in the layout process. The goal of this work is to provide a general framework through which this additional information, *ancillary information*, can be incorporated into and exploited by the

layout process. There are important differences in the kinds of information available. As a result, we have designed a very general system which should accommodate these different classes of information, from a wide variety of sources, within the objective function used during the layout process.

The ancillary information available to aid in the fragment assembly process suffers from the same problems common to most of the experimental biological data: ambiguities or errors in the data, conflicting assertions, varying reliability of data, etc. Therefore, the information can not be introduced simply as absolute constraints on the optimization; this would lead to no solution in any case with conflicting information. Instead, we treat these data items as assertions, allowing them to each contribute independently to the objective function used for the optimization process. The assertions compete with each other, as opposed to exclusively constraining the search space to one assertion or another. Additionally, we have defined a system for which objective function customization to accommodate a new kind of ancillary information will draw upon a library of primitive assertions and corresponding objective function components rather than requiring development of new functions specific to that information.

Our conceptual framework can be viewed as a logical system whose universe consists of objects and sets of objects, assertions about objects, and assertions about relationships among objects. For the sequencing process, objects are fragments, parent sequences, layouts (a *contig* is a layout in which no gaps are present), etc. A contig can be viewed both as an individual object and as an ordered set of objects — the component fragments of the contigs are the members of the set.

The information used by the assembly process is represented as assertions about objects. The framework includes different classes of assertions that characterize the manner in which an assertion can be incorporated into the objective function. Incorporating a new kind of ancillary information into this layout process requires mapping the information into one or more of the classes of assertions defined within the framework and potentially introducing new objects into the system over which these new assertions are then defined. This framework allows the objective function to be built incrementally from the components specified by the different classes of assertions. Each of the components can be weighted separately, altering the relative influence each factor has on the overall objective function. In general terms, the overall objective function, F_{total}, is defined for a particular layout l in terms of the component objective functions, F_i, corresponding to different sources or kinds of information, i,

$$F_{total}(l) = \sum_i w_i * F_i(l) \qquad (2)$$

These weights, w_i, can be set by empirical approaches or by a more formal training scheme (the latter being possible when kinds and sources of ancillary information are fixed for a series of data sets). Though we focus here on integrating ancillary assertions into optimization of layouts, some kinds of ancillary information would best be integrated into other steps (e.g., overlap strength determination) in the overall assembly.

Generic Object:

object_info(type, ID, label)
object_length(value,method)
{element_info(type, ID, label)}
span_characteristics(type, start, stop, value)

Set of Objects:

object_info(type, ID, label)
cardinality
span_characteristics(type, start, stop, value)
set_characteristics(type, frequency, value)
{object_ID}

Table 1: *Object definitions.*

In the traditional scenario where layout optimization is based only on overlap strengths determined by sequence comparison, there is a single F_i, F_{seq}, so Eq. 2 reduces to $F_{total} = F_{seq}$ (see Eq. 1).

Definition of framework

The framework for ancillary data includes object specifications and relation specifications. It is sometimes useful to treat a collection of objects, such as a set of fragments, as an object itself. Therefore, the framework includes a generic definition for objects as well as definitions for ordered and unordered sets of objects. Assertions capture the specific information about these objects. These assertions take the form of relations among objects.

Table 1 includes the generic definition of objects and sets of objects. Each object has a standard set of identifying information: an identifier, a type specifier (fragment, clone, etc.), and a label. A length can be associated with an object along with the method used to determine the length (the reliability of length information depends on how the length is determined). Components of objects are specified in the element set for the object. Finally, span_characteristics represent information that might hold over some portion of the object. For example, if the object is a fragment, the fact that the first 1000 bases are GC-rich would be represented using the span characteristic. For object sets, there is additional information for the set cardinality, whether or not the set has an ordering over it, the objects which make up the set, and any characteristics that hold for the set.

Table 2 includes an initial list of the types of as-

Spatial Relations:	Assertion Schemas:
contains(x,y)	larger_PROP(x,y,[i])
contained_in(x,y)	smaller_PROP(x,y,[i])
left_overlap(x,y,[i])	equal_PROP(x,y,[i])
left_of(x,y,[i])	not_equal_PROP(x,y,[i])
right_overlap(x,y,[i])	approx_PROP(x,y,[i])
right_of(x,y,[i])	PROP1_LOP_PROP2(x,y,[i])
offset(x,y,[i])	

Table 2: *Definition of object relations.* x and y are objects, PROP is some property or quantity (e.g., strand orientation), and LOP is a logical operator.

sertion classes required to capture biological sequencing data. Spatial relationships among fragments are prevalent in biological data, so these are enumerated separately. We then include some assertion schemas that should capture other sources of information. For example, a statement that the GC content of fragment x is 10% greater than fragment y could be stated as: larger_GC%(x,y,10). The assertion list is still under development; however, for a wide variety of ancillary information that we have identified, the list is adequate. The challenge remains to design, for each class of assertion, a mechanism within which to incorporate the assertion into the objective function and to incorporate more classes of data into the framework to further analyze its completeness.

Examples

We present three contexts where ancillary information is available and can be translated into assertions: primer-directed walking, end-sequence screening, and mapping clusters. A schematic view of these three sequencing approaches and the associated ancillary information is presented in Fig. 1. For the last case, mapping clusters, we have implemented the assertion scheme and present results for a hypothetical sequence data set, comparing the success of assembly with and without use of the ancillary information.

Primer-directed walking. Though the vast majority of sequence data available to date has been generated predominantly with random (*shotgun*) methods, directed methods have often been used to fill in gaps left over by the random approach. More recently, directed methodologies have been explored as a possible substitute for random methods altogether, not only because of their effectiveness in gap closure but also because of the reduced experimental redundancy and elimination of the need for computing the overall layout of the fragments.

One method of directed sequencing is primer-directed walking (see, for example, (Kieleczawa, Dunn, & Studier 1992)). In this technique, a first segment of the parent is sequenced and analyzed. Then, a sequencing primer is chosen that is contained by and close to the end of the first sequence. The primer is used as an anchor point from which to begin sequencing again, with the aim of extending the new sequence beyond the end of the first sequencing run. While the task of assembling these data is not as computationally complex as that required for shotgun sequencing, computational assistance is still desirable. In addition to the fragment overlap information generated by comparing the sequences, we have ancillary information corresponding to the link between the fragment from which the primer was selected and the fragment resulting from sequencing beginning at that primer. This can be translated into assertions about the relative offset and strandedness between any these pair of fragments (see Fig. 1b): the position of the primer gives an approximate offset of the second fragment relative to the first, and the known orientation of the primer gives the strandedness of the second fragment relative to the first.

We then instantiate the overall objective function according to Eq. 2,

$$F_{total} = (w_{seq} * F_{seq}) + (w_{offset} * F_{offset}) + (w_{strand} * F_{strand}) \quad (3)$$

where F_{seq} is given in Eq. 1. F_{offset} is determined as,

$$F_{offset} = \sum_i \sum_j |offset(i,j)_{anc} - offset(i,j)_{lay}| \quad (4)$$

for pairs of fragments, i, j, included in the input ancillary assertion list. The *anc* and *lay* subscripts denote data supplied as an input ancillary assertion and data generated by analysis of an output layout, respectively.[1] F_{strand} is determined as,

$$F_{strand} = \sum_i \sum_j strand(i,j) \quad (5)$$

where

$$strand(i,j) = \begin{cases} 0 & \text{if } equal\_str(i,j)_{anc} \text{ and } equal\_str(i,j)_{lay} \\ 0 & \text{if } not\_equal\_str(i,j)_{anc} \text{ and } not\_equal\_str(i,j)_{lay} \\ 1 & \text{if neither } equal\_str(i,j)_{anc} \text{ nor } not\_equal\_str(i,j)_{anc} \text{ can be asserted} \\ 2 & \text{if otherwise} \end{cases} \quad (6)$$

for pairs of fragments, i, j, included in the input ancillary assertion list. Minimizing this objective function drives toward solutions for which fragments will be offset and strand-related as indicated by the input ancillary assertions.

[1] $offset(i,j)_{lay}$ can be calculated in several different ways: some requiring a preliminary layout, some requiring estimates of overlaps, etc. A discussion of this computation and its implications for layout strategies overall is beyond the scope of this paper.

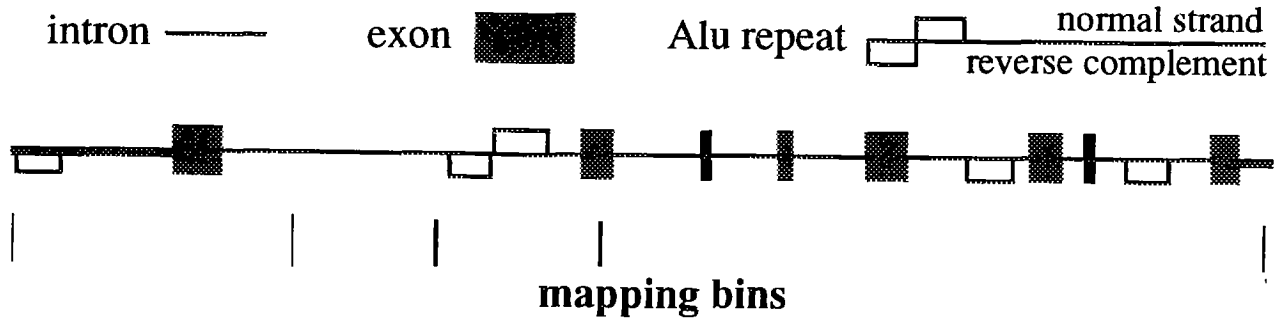

Figure 1: *Schematic representation of sequencing strategies.* Solid lines represent DNA sequences; dotted lines represent ancillary information linking the sequences. (a) Parent (double-stranded) DNA sequence. (b) Primer-directed walking. (c) End-sequence screening. (d) Mapped clusters.

End-sequence screening. Another sequencing strategy currently being explored as an improvement on shotgun sequencing is that of end-sequence screening (see, for example (Chen, Schlessinger, & Kere 1993; Smith *et al.* 1994)). A specific example is Ordered Shotgun Sequencing (OSS), where sequencing of a large parent DNA begins by breaking the parent down into smaller fragments, each several thousand bases long, that are more amenable to pure shotgun sequencing (Chen, Schlessinger, & Kere 1993). These smaller fragments' ends are sequenced, yielding several hundred bases of sequence information (in opposing strand orientation) for each end of each fragment. The approximate length of each fragment is also determined. This information can be used to identify a minimal spanning set of the fragments that will be completely sequenced, reducing the number of fragments which must be individually sequenced, and increasing the efficiency of generating a parent sequence.

The identification of the minimal spanning set relies on aligning the fragments' end sequences, and using the strand and fragment length information to develop a layout for the implied full-size fragments (see Fig. 1c). As one relies on the primary sequence data, along with ancillary information on pairwise sequence fragment offsets and strand-relatedness, to complete the layout, Eq. 3 applies equally well.

Mapped clusters. In some cases, fragments may have been localized to a sub-region of the parent DNA as a result of mapping of the sub-regions to the parent (see Fig. 1d). This can be viewed as supplying non-ordered cluster information corresponding to identification of the sub-set of fragments with a particular sub-region of the parent. Thus, we assume one or more subsets of an input set of sequence fragments have additionally been assigned to distinct clusters. (Below, we model unique cluster assignments for the case where, for example, the linking of fragments to sub-regions does not bridge across sub-regions boundaries. It would also be desirable to model the case where cluster assignments could bridge these boundaries, leading to some fragments being assigned to multiple clusters). In this case, for a layout with ordinal assignments, i, for each fragment in a set of N fragments,

$$F_{total} = (w_{seq} * F_{seq}) + (w_{cluster} * F_{cluster}), \quad (7)$$

$$F_{cluster} = \sum_{i=1}^{N-1} cluster(i,\ i+1), \quad (8)$$

and

$$cluster(i,j) = \begin{cases} 0 & \text{if } equal\_cID(i,j)_{anc} \\ 1 & \text{if neither } equal\_cID(i,j)_{anc} \\ & \text{nor } not\_equal\_cID(i,j)_{anc} \\ & \text{is asserted} \\ 2 & \text{if otherwise} \end{cases} \quad (9)$$

This composite objective function was tested on assembly of an artificial set of 177 sequence fragments drawn by **genfrag** (Engle & Burks 1993; 1994) from a known 8815 base parent sequence, HUMDKERB (Krauss & Franke 1990); the parent sequence features, and corresponding assigned clustering bins, are shown in Fig. 2.

Table 3 summarizes the results of several runs on a sample data set. Half of these runs use only sequence overlap information for the assembly optimization (designated with an A), while the other half additionally include ancillary (mapping cluster) information (designated with a B). Cluster IDs were assigned to each fragment in the input set as follows: the parent sequence was randomly divided into sections, and fragments assigned to one of six bins according to their location on the parent sequence (see Fig. 2). The results indicate that while the ancillary data does not make a major difference in all cases, it does do so a significant percentage of the time and it's results are much more consistent. Runs 1A and 1B provide an illustration: while both solutions resolved to a single contig, the finished consensus sequence of Run 1A matched the parent much less well and accounted for only 52% of the parent sequence. This solution reflects the mis-alignment of many fragments in the optimized layout, probably due to the presence of Alu repeats

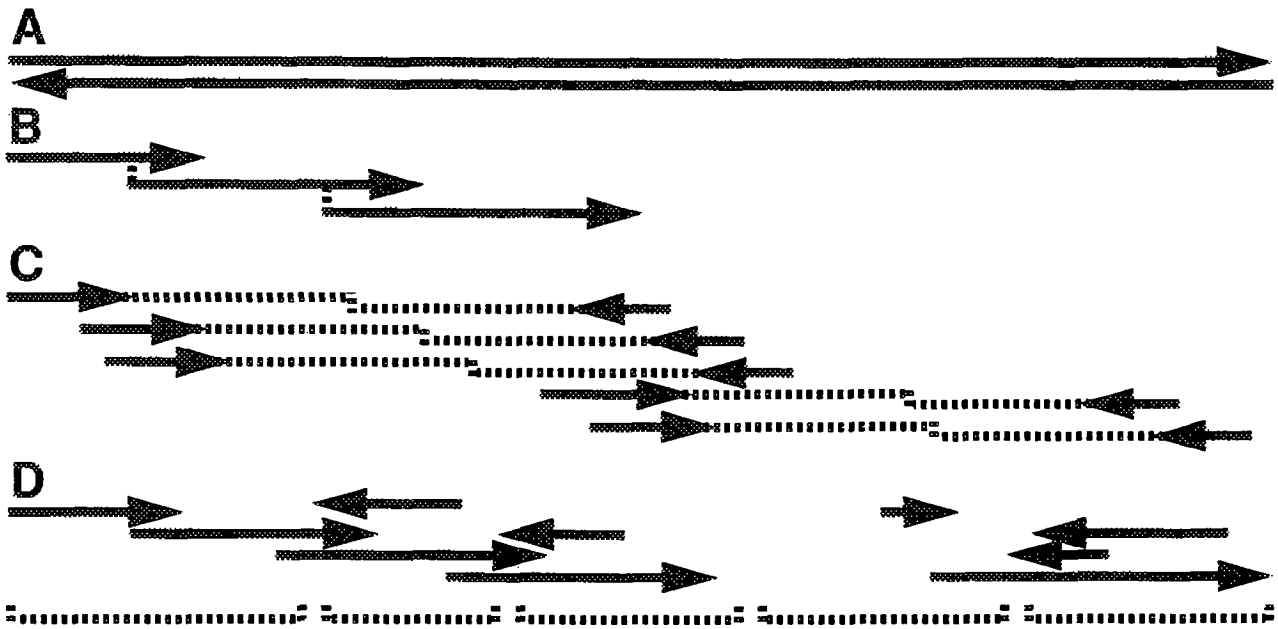

Figure 2: *Schematic representation of DNA sequence containing repeats.*

(and, in particular, the presence of an Alu at the extreme end of the parent sequence). The composite objective function generates a solution that provides a very good finished sequence although this consensus sequence is 12 bases longer than the parent (as shown in Table 3). This increase in precision and reliability only costs about 20% in execution time (13:36 to 17:32 in cpu time for a typical run).

Run	Ctgs	% Cvg	% Match
1A	1	64.9	52.2
1B	1	100.1	99.9
2A	1	96.6	96.0
2B	1	100.1	99.8
3A	1	100.2	99.9
3B	1	100.2	99.4
4A	1	100.0	99.9
4B	1	100.0	99.4
5A	2	95.9	83.4
5B	1	99.9	99.7

Table 3: *Results of integrating ancillary mapping cluster information.* Each numbered pair of runs corresponds to a different random seed; within each pair, the A run corresponds to assembly based only on pairwise sequence overlap information, while the B run additionally uses ancillary clustering information. *Ctgs* indicate the number of contigs in the final solution; *Cvg* is the percent of the parent that is covered by the final solution; *Match* is the percent of the parent that is correctly matched by the final solution.

The w_i values for these runs were set so that the $w_i * F_i$ were of approximately the same magnitude (given initial values of F_i), based on the intuitive notion of exploring the effect of equal contributions from the two kinds of data, and within a range where minor changes in w_i did not appear to have a major effect on the output solutions. As noted above, it would be desirable to develop a theoretical basis for assigning these values.

Discussion

We have presented an exploratory description of an approach to integrating competing ancillary assertions in genome assembly optimization. Though our results are preliminary, they are consistent with our goals. Two apparently different kinds of information (the first and second examples in the section above) were mapped to the same class of assertions and resulting composite objective function. The third, implemented example demonstrated a situation where ancillary information drives the assembly optimization to a better result than otherwise. We intend to begin a more systematic collection and translation of different kinds of ancillary information, to implement more of the basic component objective functions, and to begin testing this approach on experimental data.

Alternative approaches

One technique frequently explored for dealing with different classes of information is to translate the additional information into constraints and perform some form of constraint propagation to yield a result. This approach has been implemented for optimizing the or-

dering of genetic maps (Letovsky & Berlyn 1992) and physical maps (Soderlund & Burks 1994); more recently, similar strategies for sequence assembly have been described (Burcham 1994; Jain, Larson, & Myers 1993). While this technique has been used successfully in many different areas of optimization, constraint propagation requires any solution to satisfy all constraints. This is not possible if constraints are contradictory. In addition, it is not always possible to precisely quantify a particular constraint; length information is imprecise, as an example. One can use intervals to address the ambiguity (Letovsky & Berlyn 1992), but this requires a precise estimate of the possible error.

Another potential approach to this problem would be to use bayesian techniques, which are useful in balancing related probabilistic events [see for example (Press 1989)]. However, it is not clear how to map several of the important assertions into this framework, given the arbitrariness required for the selection of the probability distributions.

Potential for generalizing the current approach

Although we have implemented our strategy in the context of a particular model for assembling sequences with stochastic searches, we believe that the conceptual framework will be applicable to other assembly strategies. For example, in greedy strategies – rather than casting the ancillary assertions in terms of globally-summed objective functions – the summation of component objective functions would have to occur at the level of pairwise fragment relations. These summed pairwise assessments could then be used in the usual way to drive the greedy construction. The results of an implementation in a greedy context would be expected to differ from those presented here.

As shown above, the current approach is not limited to a particular sequencing strategy. Similarly, it should be applicable to mapping strategies based on a larger scale than sequencing (e.g., building restriction maps, or physical mapping based on clone fingerprinting). These problems can often be cast in very similar terms, and are certainly at least as complicated due to the presence of experimental error as well as multiple sources and kinds of input information (see, for example (Fickett & Cinkosky 1993; Graves 1993; Letovsky & Berlyn 1992; Pratt & Dix 1993; Skiena & Sundaram 1993; Soderlund, Torney, & Burks 1993; Stam 1993; Soderlund & Burks 1994)).

Using ancillary assertions in data mining contexts involving sequence or higher-level, mapping data might also be greatly advantageous. An example would be the assembly of published sequences that are likely to overlap with other published sequences, but stored in separate pieces, in different places, and with different annotation standards.

Finally, the class of problems (genome map assembly) we have addressed here can be abstracted as the generation of interleaved tilings of overlapping one-dimensional objects. The possibility exists of applying our approach to integration of competing ancillary assertions in similarly abstracted contexts such as the alignment of geological core samples or the alignment of overlapping tasks in manufacturing process control.

Acknowledgements

We appreciate insights that arose in conversations with T. Burcham, E. Chen, W. Fulkerson, J. Gatewood, T. Hunkapiller, S. Letovsky, J. Mills, G. Moody, C. Soderlund, J. Stewart, and L. Zuo; and also appreciate several useful suggestions from the referees. This work was funded in part by the Los Alamos High Performance Computing Center and the Los Alamos Computational Testbed for Industry under the auspices of the Dept. of Energy. R.P. was supported by a LANL Director's Postdoctoral Fellowship. Part of this work was done during a workshop at the Aspen Center for Physics funded by the National Science Foundation.

References

Burcham, T. 1994. Personal communication.

Burks, C.; Cinkosky, M.; Gilna, P.; Hayden, J.-H.; Keen, G.; Kelly, M.; Kristofferson, D.; Fischer, W.; and Lawrence, J. 1992. GenBank. *Nucl. Acids. Res.* 20:2065–2069.

Burks, C.; Engle, M.; Lowenstein, M.; Parsons, R.; and Soderlund, C. 1993. Stochastic optimization tools for DNA assembly: integration of physical map and sequence data. Poster presented at Genome Sequencing and Analysis Conference V, Hilton Head, NC.

Burks, C.; Engle, M.; Forrest, S.; Parsons, R.; Soderlund, C.; and Stolorz, P. 1994. Stochastic optimization tools for genomic sequence assembly. In Venter, J. C., ed., *Automated DNA Sequencing and Analysis.* Academic Press. 249–259.

Chen, E.; Schlessinger, D.; and Kere, J. 1993. Ordered shotgun sequencing (OSS), a strategy for integrated mapping and sequencing of YAC clones. *Genomics* 17:651–656.

Churchill, G.; Burks, C.; Eggert, M.; Engle, M.; and Waterman, M. 1993. Assembling DNA sequence fragments by shuffling and simulated annealing. Technical Report LA-UR-93-2287, Los Alamos National Laboratory, Los Alamos, NM.

Dear, S., and Staden, R. 1991. A sequence assembly and editing program for efficient management of large projects. *Nucl. Acids Res.* 19:3907–3911.

Engle, M., and Burks, C. 1993. Artificially generated data sets for testing DNA fragment assembly algorithms. *Genomics* 16:286–288.

Engle, M., and Burks, C. 1994. GenFrag 2.1: new features for more robust fragment assembly benchmarks. *Comp. Applic. Biosci.* In press.

Engle, M.; Parsons, R.; and Burks, C. 1994. MFA tool for rapid alignment of multiple overhanging sequences. Unpublished software.

Fickett, J., and Cinkosky, M. 1993. A genetic algorithm for assembling chromosome physical maps. In Lim, H.; Fickett, J.; Cantor, C. R.; and Robbins, R., eds., *Proceedings of the Second International Conference on Bioinformatics, Supercomputing, and Complex Genome Analysis.* World Scientific. 273–285.

Graves, M. 1993. Integrating order and distance relationships from heterogenous maps. In Hunter, T.; Searls, D.; and Shavlik, J., eds., *Proceedings: First International Conference on Intelligent Systems for Molecular Biology,* 154–162. Menlo Park, CA: AAAI Press.

Huang, X. 1992. A contig assembly program based on sensitive detection of fragment overlaps. *Genomics* 14:18–25.

Hunkapiller, T.; Kaiser, R.; Koop, B.; and Hood, L. 1991. Large-scale and automated DNA sequence determination. *Science* 254:59–67.

Jain, M.; Larson, S.; and Myers, G. 1993. A new software kernel for fragment assembly. Poster presented at Genome Sequencing and Analysis Conference V, Hilton Head, NC.

Kececioglu, J., and Myers, E. 1989. A procedural interface for a fragment assembly tool. Technical Report TR-89-5, Depart of Computer Science, University of Arizona, Tucson, AZ.

Kececioglu, J. 1991. *Exact and approximation algorithms for DNA sequence reconstruction.* Ph.D. Dissertation, University of Arizona, Tucson, AZ. TR 91-26, Department of Computer Science.

Kieleczawa, J.; Dunn, J.; and Studier, F. 1992. DNA sequencing by primer walking with strings of contiguous hexamers. *Science* 258:1787–1791.

Krauss, S., and Franke, W. 1990. Organization and sequence of the human gene encoding cytockeratin 8. *Gene* 86:241–249.

Letovsky, S., and Berlyn, M. 1992. CPROP: A rule-based program for constructing genetic maps. *Genomics* 12:435–446.

Myers, E. 1994. Advances in sequence assembly. In Venter, J., ed., *Automated DNA Sequencing and Analysis.* London, England: Academic Press. 231–238.

Parsons, R.; Forrest, S.; and Burks, C. 1993. Genetic algorithms for DNA sequence assembly. In Hunter, T.; Searls, D.; and Shavlik, J., eds., *Proceedings: First International Conference on Intelligent Systems for Molecular Biology,* 310–318. Menlo Park, CA: AAAI Press.

Parsons, R.; Forrest, S.; and Burks, C. 1994. Genetic operators for the DNA fragment assembly problem. *Machine Learning.* Accepted for publication.

Pearson, W. 1993. The FASTA program package. Software manual, U. Virginia, Charlottesville, VA.

Pratt, D., and Dix, T. 1993. Construction of restriction maps using a genetic algorithm. In Mudge, T.; Milutinovic, V.; and Hunter, L., eds., *Proceedings of the Twenty-Sixth Hawaii International Conference on System Sciences. Vol. I: Systems Architecture and Biotechnology,* 756–762. Los Alamitos, CA: IEEE Computer Society Press.

Press, S. J. 1989. *Bayesian Statistics: Principles, Models and Applications.* Wiley, New York.

Skiena, S., and Sundaram, G. 1993. A partial digest approach to restriction site mapping. In Hunter, T.; Searls, D.; and Shavlik, J., eds., *Proceedings: First International Conference on Intelligent Systems for Molecular Biology,* 362–370. Menlo Park, CA: AAAI Press.

Smith, M.; Holmsen, A.; Wei, Y.; Peterson, M.; and Evans, G. 1994. Genomic sequence sampling: a strategy for high resolution sequence-based physical mapping of complex genomes. *Nat. Genet.* In press.

Soderlund, C., and Burks, C. 1994. GRAM and genfragII: simulating and solving the single-digest partial restriction map problem. *Comp. Applic. Biosci.* In press.

Soderlund, C.; Torney, D.; and Burks, C. 1993. Calculating shared fragments for the single digest problem. In Mudge, T.; Milutinovic, V.; and Hunter, L., eds., *Proceedings of the Twenty-Sixth Hawaii International Conference on System Sciences. Vol. I: Systems Architecture and Biotechnology,* 620–629. Los Alamitos, CA: IEEE Computer Society Press.

Stam, P. 1993. Construction of integrated genetic linkage maps by means of a new computer package: JoinMap. *Plant J.* 739–744.

Venter, J., ed. 1994. *Automated DNA Sequencing and Analysis.* London, England: Academic Press.

Wilson, R.; Ainscough, R.; Anderson, K.; Baynes, C.; Berks, M.; Bonfield, J.; Burton, J.; Connell, M.; Copsey, T.; Cooper, J.; Coulson, A.; Craxton, M.; Dear, S.; Du, Z.; Durbin, R.; Favello, A.; Fraser, A.; Fulton, L.; Gardner, A.; Green, P.; Hawkins, T.; Hillier, L.; Jier, M.; Johnston, L.; Jones, M.; Kershaw, J.; Kirsten, J.; Laisster, N.; Latreille, P.; Lightning, J.; Lloyd, C.; Mortimore, B.; O'Callaghan, M.; Parsons, J.; Percy, C.; Rifken, L.; Roopra, A.; Saunders, D.; Shownkeen, R.; Sims, M.; Smaldon, N.; Smith, A.; Smith, M.; Sonnhammer, E.; Staden, R.; Sulston, J.; Thierry-Mieg, J.; Thomas, K.; Vaudin, M.; Vaughan, K.; Waterson, R.; Watson, A.; Weinstock, L.; Wilkinson-Sproat, J.; and Wohldman, P. 1994. 2.2 Mb of contiguous nucleotide sequence from chromosome III of *C. elegans.* *Nature* 368:32–38.