Engineering in Genomics

Martin Farach, Editor

DNA Sequence Assembly

by Christian Burks
Los Alamos National Laboratory

The Human Genome Project is developing a complete representation of the information underlying our genetic make-up, as well as the genetic make-up of several other species that are either important for comparative scientific understanding, relied on for progress in human medicine, or of direct industrial or agricultural interest [1,2]. This information, encoded linearly along the genomic DNA polymer, will be used as a platform to rapidly expand our ability to characterize and understand human disease and infection, design and develop preventative and therapeutic medicines, develop improved nutritional sources, and employ microbial tools in environmental and other applications. It is widely held that DNA sequencing throughput will have to be increased by orders of magnitude to complete the task in the time frame of 15 years that was laid out for the Human Genome Project, and that such dramatic increases will rely in large part on automating the several experimental and interpretive steps involved in DNA sequencing.

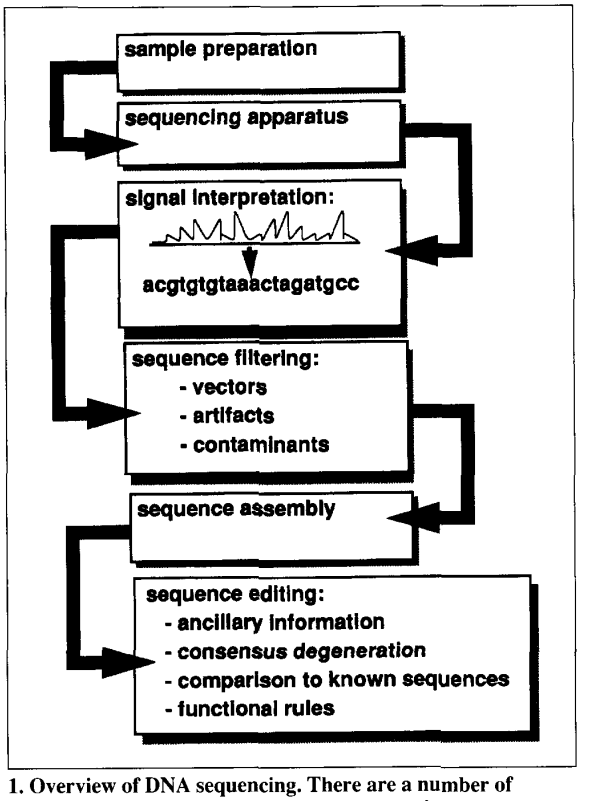
DNA sequencing (Fig. 1) is the highest-resolution approach to genome mapping. It usually involves pieces of DNA (e.g., chromosomes, or sub-chromosomal regions to which a disease gene of interest has previously been mapped) that are much too long to be successfully analyzed— as long, intact molecules— with currently available biochemical methods. Thus, one breaks the long piece of DNA into shorter pieces of DNA that are amenable to current experimental methods, sequences the individual pieces, and then uses the fact that overlapping pieces will have common subsequences to stitch the sequences of individual pieces into a representation of the original, long piece. We focus here on the challenges associated with developing algorithms and software tools for automating the assembly of the pieces to reconstruct the original sequence.

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 monomeric bases of which the DNA polymer is composed. A piece, or fragment, corresponds in our context to a sub-string of 100-1000 bases. Overlap strength and offset relationships between pairs of fragments, used to drive the assembly of the fragments (Fig. 2) into a global layout, is based on comparison of character strings. The output generated by sequencing represents a consensus on the order of 1000-1,000,000 bases long, generated by voting in aligned columns of bases resulting from the layout.

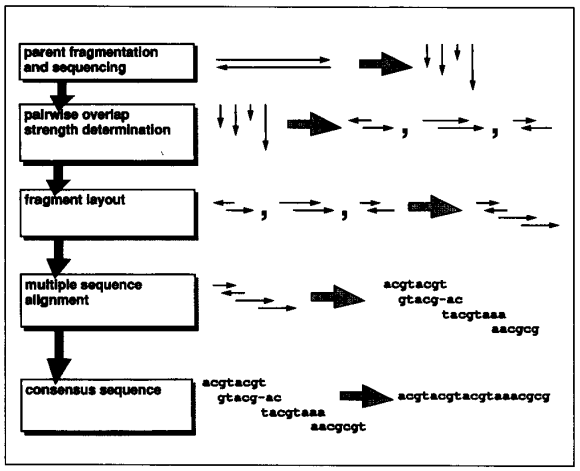
Recently, several groups have published the results of large-scale sequencing projects generating from tens of thousands to millions of contiguous bases (e.g., [3]). In contrast with previously published sequences of comparable size, these efforts are noteworthy because of their conception and implementation as short-term, globally-comprehensive sequencing projects. The size of these projects, and the even more ambitious goals of the Human Genome Project, demand a particular emphasis on developing large-scale, high-throughput DNA sequencing [4,5].

Complications

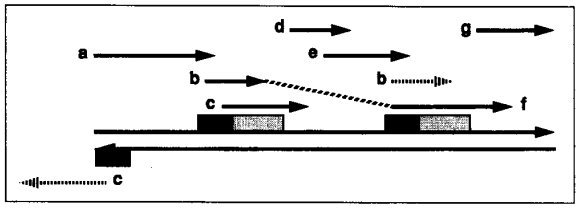
The limitation of direct experimental determination to stretches of sequence that are short relative to the target, parent sequences has meant that for a large project, involving hundreds or thousands of sequence fragments, the computation of an optimal layout requires alternatives to the systematic exploration of all possible assemblies. A number of other factors further complicate the computational complexity of sequence assembly. Though sequencing both of the complementary, anti-parallel strands of the DNA double helix helps minimize errors arising in the determination of base identities, or base calling, it also leads to the need to assign and track strand sense through the assembly 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



1. Overview of DNA sequencing. There are a number of opportunities for either instrumental automation or computational tools (or both) at each of these steps.



2. Overview of DNA sequence assembly. This is one of several possible stepwise representations of the process of fragment assembly. The solid lines represent sequence fragments, and their arrows indicate orientation along the DNA polymer. The character strings represent higher-resolution views of the fragments.



3. Problems arising with repetitive DNA sequences. The solid, arrowed lines represent fragments in their true layout positions (the two long lines represent the parent, double-stranded DNA sequence). The presence of repeats, indicated by the shaded boxes along one or the other strand of the parent sequence, can lead to incorrect layouts, indicated with stippled lines. In one case, fragment *b* could be aligned in the wrong region; in a second case, fragment *f* could be aligned with fragment *b*, leading to a contracted layout; and the final case, fragment *c* could be aligned in the opposite orientation, leading to an incorrectly divergent layout.

information (Fig. 3); this problem is exacerbated by higher rates of conservation among and larger repeat units in a repeat family. Finally, the input fragment sequences usually include experimental ambiguities or errors that affect assessment of pairwise overlap strength and subsequent detailed alignments.

There are a number of sources of experimental error, including the substitution, insertion, and deletion of bases in determining sequence fragments; the ambiguous (uncertain) determination of bases; and the presence of artifact sequences arising either from rearrangement of the target, parent sequence or inclusion of unrelated sources of DNA. Furthermore, the rates and distributions of errors can vary with source (e.g., different protocols, technicians, and sources of material).

There are a number of compensating strategies for addressing these problems. Improved chemistry and instrumentation, either to decrease error rates or to increase the lengths of the directly-determined sequence fragments, reduces the complexity and uncertainty of assembly. Increased depth of coverage (with fragments) of a given target region is another approach to decreasing the uncertainty of the result. Ancillary information about the the relationships among fragments can augment or be made to over-ride the primary sequence comparison data, constraining the placement of fragments in the assembly layout. This information can arise from related, independent experiments (e.g., mapping data on the parent sequence) or as a result of a particular sequencing strategy (e.g., where the sequence of the end of one fragment is used to formulate a chemical template for determining the sequence of an adjacent, overlapping fragment). Prior knowledge of the sequence of repeat family members (some of which are well-documented in a given species) allows one to down-weight the effect of those sequences on the assembly. There has also been considerable focus on developing computational tools that have the potential of speeding-up and increasing the efficiency of assembly of large fragment sets, and on conceptual and run-time improvements in the algorithms on which these tools rely.

Assembly Algorithms

The most prevalent algorithmic approach to assembly has been the greedy construction of a single or a few solutions (e.g., [6,7]), where one builds up the layout (a layout composed of entirely interconnected fragments is called a contig by sequential addition, one fragment at time, based on their pairwise overlap strengths. More global approaches have also been explored. We have implemented stochastic search strategies such as relaxation, simulated annealing, and genetic algorithms (e.g., [8,9]). Others have implemented a tiered cluster approach (e.g., [10]), or rapid approximations to exact, global constructions based on formulating the assembly in terms of finding the shortest common superstring (e.g., [11]). A number of these these algorithms have recently been reviewed by Myers [12].

Lacking from the software packages widely-used to date is the use of and dependence on confidence levels associated with individual bases in the input strings, in large part because the experimental community has not traditionally preserved these data. There is now considerable interest both in translating the raw sequencing data into base calls and their associated confidence values, and in using these more highly articulated data in the various string-processing algorithms associated with assembly.

Software Engineering

There are several issues that arise, and that have not been completely settled, in the process of defining, implementing, and evaluating algorithms for assembly. First, experimentalists often prefer, and the data often demand, a strategy based on computer assistance rather than purely automated assembly (both because definitions of what constitutes an optimal assembly are varied or imprecise, and because the error rates lead to more correct solutions being less optimal). That is, the final sequence output depends on manual intervention and editing at one or more stages. Thus, when an end-user reports that they have used a particular tool to assemble data, it is often difficult to assess how much of the result (and whether or not it was satisfactory) was due to the automated steps, and how much due to the manual intervention. In addition, formulations of the assembly problem have varied considerably, particularly with respect to the degree of sequential modularity of the problem; increased modularity is often reflected in separate algorithms for each step, so that there is not really a single assembly algorithm to evaluate.

There is no standard measure of the quality of an output assembly. The speed of attaining a result is always important, but the quality of the result has variously been quantified as lower number of output contigs, higher percent coverage of the parent sequence, less ambiguity in the columns of aligned bases contributing to the output consensus sequence, and higher percent

match of the output consensus sequence to the parent sequence in the regions that are covered. Some of these measures can only be calculated when using artificial data sets where the parent sequence is known *a priori*, or by holding up a result from another assembly of an experimental data set as the true result. Finally, there has been little standardization of benchmark data sets on which to test assembly algorithms.

Benchmark Data Sets

What data sets are to be used to test new algorithms and the tools in which they are imbedded? Obviously, the ultimate testbed and source of benchmarks are experimental data sets, accessible either where they have been posted for that explicit purpose (e.g., [13]) or by wandering down the (electronic) hall to the nearest sequencing laboratory and asking for sample data sets. However, these experimental data sets usually represent a very sparse sampling of the solution space of data set parameters, making it difficult to isolate and assess the impact of any single data parameter on the assembly algorithm being tested. In this context, being able to generate artificial data sets, by computationally fragmenting known sequences, can be very useful. For example, we have developed a tool [14] that provides systematic, independent variation of fragment set parameters, including range and mean of fragment lengths, mean depth of coverage, mean error rates, error distribution along fragments, and repeat complexity of the parent sequence from which the fragments are drawn.

Ancillary Information

As noted above, several sources of ancillary information can potentially compensate for either incompleteness or errors in the primary sequence data. The traditional approach to sequence assembly relies on a well-defined algorithm to automatically generate a layout based only on the primary sequence data, usually followed by a lengthy, manual editing process to incorporate the ancillary information that the experimentalist has at hand. However, in the interest of ramping up throughput rates (and therefore reducing the amount of manual editing), it is desirable to consider ways in which the ancillary information can incorporated into the automatic assembly of layouts.

One can translate the ancillary information into constraints and perform some form of constraint propagation while generating a solution. This approach has been implemented for optimizing the ordering of genetic maps [15] and physical maps [16]. However, constraint propagation requires any solution to satisfy all constraints, which is not possible if constraints are contradictory, as is often the case in the face of errors associated with both the primary sequence data and the associated, ancillary information. In addition, it is not always possible to precisely quantify a particular constraint; for example, offset information relating a pair of fragments is often imprecise. Alternatively, one could use bayesian techniques, often used in balancing related probabilistic events. However, it is not clear how to map several of the important assertions relating fragments to one another into this framework, given the arbitrariness required for the selection of the probability distributions.

Yet another approach is to develop a general framework of assertion classes for defining and quantifying the various relationships among fragments (overlap strength of their sequences, offsets in the layout, and so on), develop a corresponding library of objective function components corresponding to these assertion classes, and to build up objective functions from these components for layout optimization based on competition among the various individual ancillary assertions [17].

Christian Burks is Program Coordinator for the Computational Biology Program at Los Alamos National Laboratory. He received a B.A. in the Great Books Program from St. Johns College and a Ph.D. in Molecular Biophysics and Biochemistry from Yale University. He then joined the Theoretical Biology and Biophysics Group at Los Alamos National Laboratory, first as a post-doctoral fellow, then as a Staff Member, and finally as Group Leader. During this period his primary focus was the GenBank DNA sequence database project, the Los Alamos component of which he led from 1987-1992. Recently, he has shifted his focus from databases to genome assembly and DNA sequence analysis problems. In his current position, he is developing an institution-wide program in computational biology. His address is: LANSCE/ER, MS K710; Los Alamos National Laboratory; Los Alamos, NM 87545 (e-mail, cb@t10.lanl.gov; fax, 505-665-3493; tel, 505-667-6683).

References

1. **Collins F, and Galas D:** A new 5-year plan for the United-States Human Genome Project. *Science*, 262, 43-46, 1993.
2. **Cooper NG, ed:** *The Human Genome Project: Deciphering the Blueprint of Heredity* , University Science Books, Mill Valley, CA, 1994.
3. **Wilson R, Ainscough R, Anderson K, Baynes C, Berks M, et al:** 2.2 Mb of contiguous nucleotide sequence from chromosome III of C. elegans. *Nature*, 368, 32-38, 1994.
4. **Hunkapiller T, Kaiser RJ, Koop BF, and Hood L:** Large-scale and automated DNA sequence determination. *Science*, 254, 59-67, 1991b.
5. **Adams MD, Fields C, and Venter JC, eds:** *Automated DNA Sequencing and Analysis*, Academic Press, New York, NY, 1994.
6. **Dear S. and Staden R:** A sequence assembly and editing program for efficient management of large projects. *Nucl. Acids Res.*, 19, 3907-3911, 1991.
7. **Huang X:** A contig assembly program based on sensitive detection of fragment overlaps. *Genomics*, 14, 18-25, 1992.
8. **Burks C, Engle ML, Forrest S, Parsons RJ, Soderlund CA, and Stolorz PE:** Stochastic optimization tools for genomic sequence assembly. In: *Automated DNA Sequencing and Analysis*, Adams M.D, Fields C, and Venter JC, eds., Academic Press, New York, pp. 249-259, 1994.
9. **Parsons R, Forrest S, and Burks C:** Genetic algorithms for DNA sequence assembly. In: *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, Hunter T, Searls D, and Shavlik J, eds, AAAI/MIT Press, Menlo Park, CA, 310-318, 1993.
10. **Gleizes A, and Henaut A:** A global approach for contig construction. *Comp. Applic. Biosci.* , 10, 401-408, 1994.
11. **Kececioglu JD:** Exact and approximation algorithms for DNA sequence reconstruction. Ph.D. Thesis (TR#91-26), Department of Computer Science, U. Arizona, Tucson, AZ (1991).
12. **Myers EW:** Advances in sequence assembly. In: *Automated DNA Sequencing and Analysis* , Adams MD, Fields C, and Ventner JC, eds., Academic Press, New York, pp. 249-259, 1994.
13. **Seto D, Koop BF, and Hood L:** An experimentally-derived data set constructed for testing large-scale DNA sequence assembly algorithms. *Genomics*, 15, 673-676, 1993.
14. **Engle ML and Burks C:** Artificially generated data sets for testing DNA sequence assembly algorithms. *Genomics*, 16, 286-288, 1993.
15. **Letovsky S and Berlyn MB:** CPROP: A rule-based program for constructing genetic maps, *Genomics*, 12, 435-446, 1992.
16. **Soderlund CA and Burks C:** GRAM and genfragII: simulating and solving the single-digest partial restriction map problem. *Comp. Applic. Biosci.*, 10, 349-358, 1994.
17. **Burks C, Parsons RJ, and Engle ML:** Integration of competing ancillary assertions in genome assembly. In: *Proceedings Second International Conference on Intelligent Systems for Molecular Biology* , Altman R, Brutlag D, Karp P, Lathrop R, and Searls D, eds, AAAI Press, Menlo Park, CA, pp. 62-69, 1994.