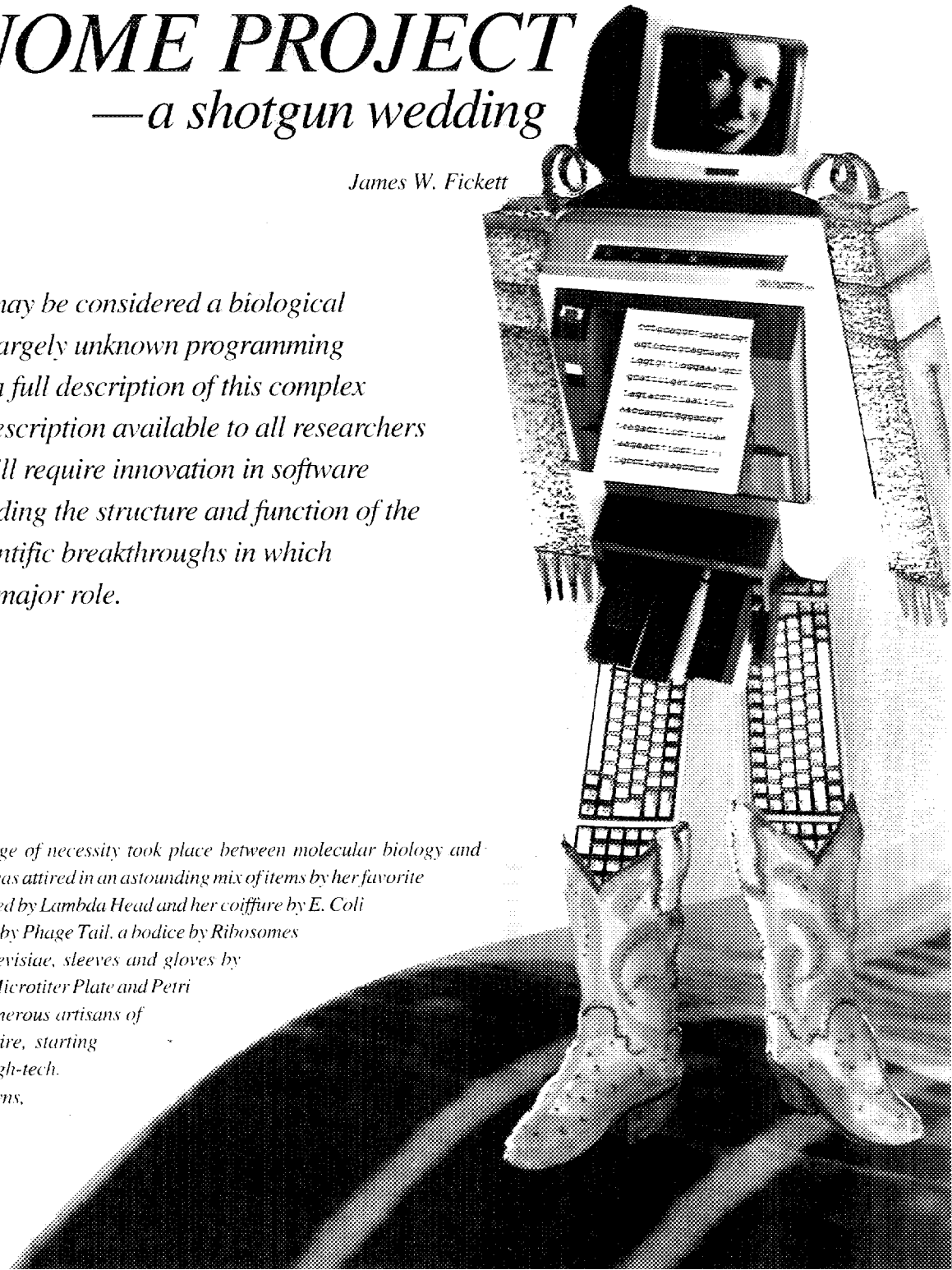
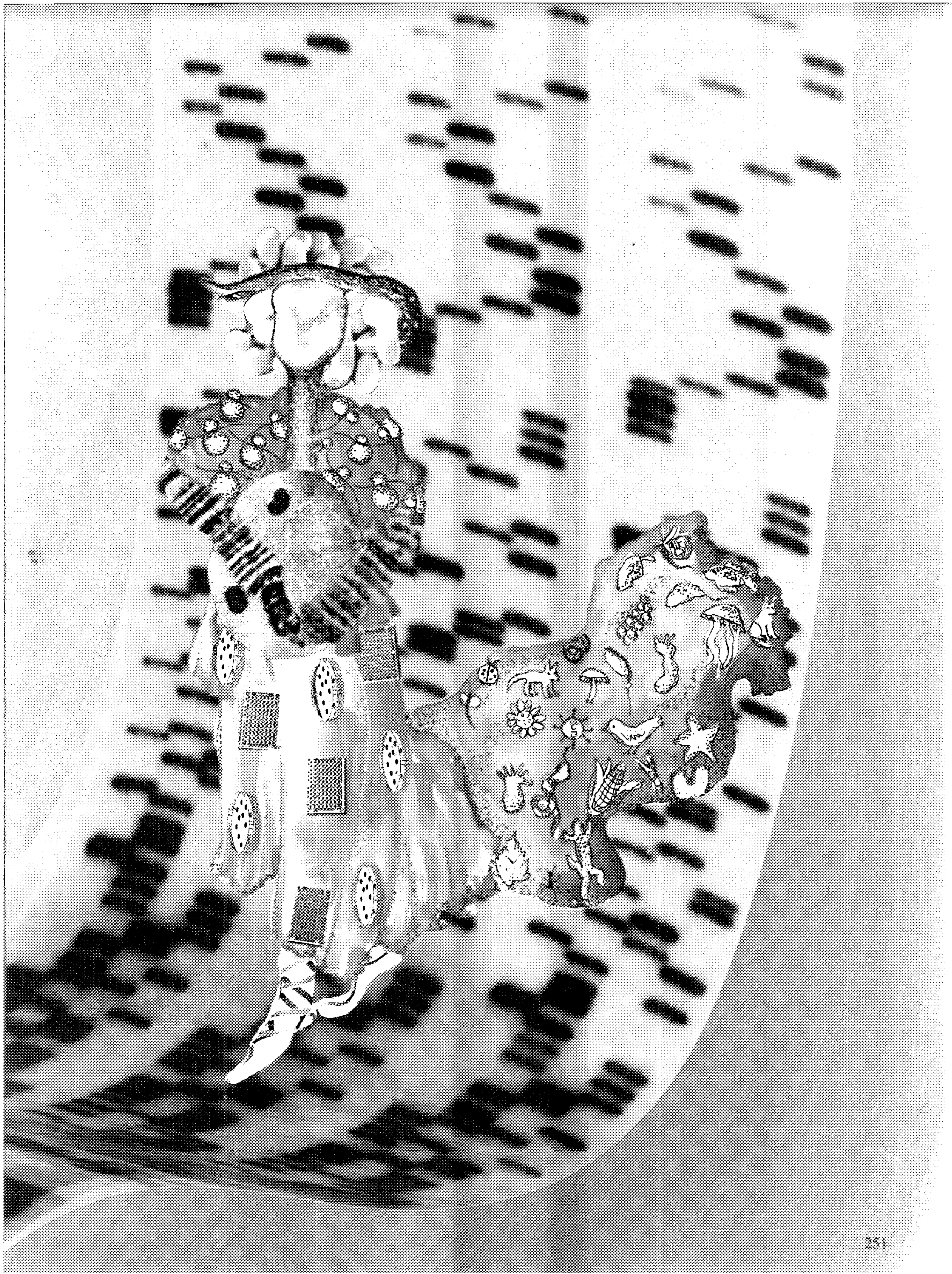
COMPUTATION *and the GENOME PROJECT*
—a shotgun wedding

James W. Fickett

The human genome may be considered a biological "program" written in a largely unknown programming language. Assembling a full description of this complex object and making the description available to all researchers via computer network will require innovation in software engineering. Understanding the structure and function of the genome will require scientific breakthroughs in which computation will play a major role.



Several years ago a marriage of necessity took place between molecular biology and computational science. The bride was attired in an astounding mix of items by her favorite designers. Her makeup was executed by Lambda Head and her coiffure by E. Coli and C. Elegans. She wore a choker by Phage Tail, a bodice by Ribosomes Unlimited, a stomacher by S. Cerevisiae, sleeves and gloves by Chromosomes to Order, a skirt by Microtiter Plate and Petri Dish, and a train fashioned by numerous artisans of Wonderful Life. The groom's attire, starting with his essential boots, is strictly high-tech. The marriage has had its ups and downs, but both partners are starting to learn from one another.



The genome is more than a blueprint

Elsewhere in this issue the nature and function of the human genome are described from a biochemical point of view. We begin by describing the genome in computational terms. Since the DNA polymer is made up of four monomer units, whose standard abbreviations are A, C, G, and T, a DNA molecule may be represented by a character string using only these four letters. The chemical monomers are called nucleotides; the strings are known as nucleotide sequences. The human genome, from this point of view, is a set of 24 character strings (representing 24 chromosomes), with a total length of a few billion letters, that is, with a size of a few gigabytes.

The genome is often called the blueprint for the species. In brief, and very roughly speaking, the genome is a concatenation of genes; each gene contains the plans for a protein; and proteins are the key building blocks of the body. (Essentially all enzymes—biological catalysts—are proteins, much of the structure of the body is protein, and the molecules that are not proteins are made by those that are.) For a description of how a gene is expressed to produce a protein see "Protein Synthesis" in "Understanding Inheritance."

The blueprint metaphor is very useful, but does break down in some respects. A blueprint for a home normally depicts only the home. But the genome, even as a blueprint, does much more. There are enzymes that read genes and make the corresponding proteins, and the genome specifies these (as if a blueprint contained drawings for hammers, nails, and workmen). There are even enzymes for rearranging the genome (as if a blueprint were to specify an independent-minded contractor).

Furthermore, the genome contains many regions that, rather than listing specifications for protein, interact with enzymes in process-control mechanisms. For example, certain enzymes known as transcription factors must bind to control regions near a gene each time that gene is used to produce a protein. Such regions are altogether outside the blueprint metaphor. So it is profitable instead to think of the genome as a program, written in a largely unknown programming language. Within the program are data arrays—the codon triplets that account for the "blueprint" parts of genes. The main program encodes a number of other related programs that act on the main one: a copier, interpreters, and rearrangers. A good part of the main program is concerned with proper communication between the main and related programs.

The goal of the Human Genome Project is an atlas

The final goal is the annotated sequence. The eventual goal of the Human Genome Project is to obtain the full nucleotide sequence of the genome, with each region annotated as to function. From the point of view of the program metaphor, this means obtaining a full, documented listing of the program.

In one sense this goal is only the culmination of a trend. It has become clear over the last two decades that almost any problem in biology can be more easily solved if the underlying genetic specification (that is, the annotated nucleotide sequence) of the relevant biochemistry is known. And because of the revolution in biotechnology, we are now able to see the genetic specification of any organism in as much detail as we wish (and can afford: the current cost of sequencing an average gene is on the order of $10,000, and isolating the relevant genetic material may well cost more). Annotated nucleotide sequences have thus accumulated at an exponentially increasing rate.

However, the Human Genome Project goes far beyond the trend of ever increasing sequence determination, for its aim is not just more sequence. In fact the hallmark of the genome project is an interest in the design and working of the genome as an organic whole.

Obtaining the full sequence and gaining an understanding of its overall organization will require many years and a significant amount of money. What will we gain that could not be had by a piecemeal approach? One example comes from the determination by a European collaboration of the full sequence of yeast chromosome III. (The human genome project includes the study of several model organisms.) One of the surprises in this sequence is that there seem to be many more genes than expected. Since the functions of most of those genes are not yet known, their discovery by other methods would have been long in coming.

On a more fundamental level, through the genome project we will learn a great deal about the programming language in which organisms are specified. The human genome is quite possibly the most complex object yet studied by science, encoding thousands of protein products which, working together in intricate combinations, manage the genetic program, build the human body from scratch, and maintain it for a

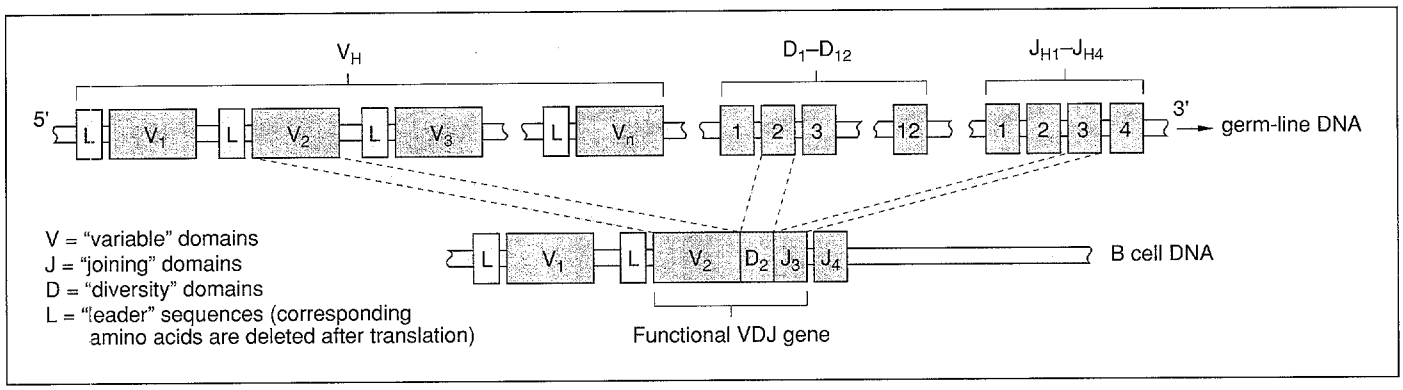


Figure 1. More Complex Genes: The Immunoglobulins

The immune system produces somewhere between a million and a hundred million different immunoglobulins. If each of these protein antibodies were encoded by a separate gene, the genome would have no room to encode anything else. In fact the immunoglobulins are specified in a tiny fraction of the genome. How this is accomplished is an excellent example of "genome programming." A typical immunoglobulin molecule is made up of four protein subunits: two identical "heavy chains" and two identical "light chains." Each of these has a "constant region" to interact with immune-system cells, and a "variable region" that is specific to a particular foreign molecule. The figure shows a schematic diagram of the genetic information corresponding to the variable region of a heavy chain. In germ-line DNA, that is, the DNA inherited from one's parents, there are several hundred V ("variable") domains, followed by twelve D ("diversity") domains, followed by four J ("joining") domains. In lymphocytes (white blood cells) this DNA is rearranged so that a particular V, D, and J region are joined to make an exon for the variable region of the heavy chain. Many thousands of different combinations are produced in different cells. In addition, the rearrangement is somewhat inaccurate, producing more variants. Also, in this region mutations are unusually common, even during the life of one cell, producing still more variation. The light chains are produced by similar mechanisms. Finally, each of the many light chains can pair with each of the many possible heavy chains, so that there are billions of possible immunoglobulins. From these the immune system duplicates and maintains those that turn out to be useful in recognizing foreign molecules.

lifetime. We now know little bits of how this complexity is orchestrated; concentrating on the big picture will teach us much more.

Second, the fully described sequence, like a geographic atlas or a star atlas, is a resource of enduring interest. In a deep sense biology, especially molecular biology, is data-driven. While physics and chemistry deal with general laws, biology, like geography and history, deals in large part with many specific cases. There are generalizations in biology but, while the generalizations of physics and chemistry are close to being exact models from which one can predict the behavior of matter, the generalizations of biology are more in the nature of analogy. They guide the intuition rather than enabling one to predict the behavior of the system. General principles in biology are frequently implemented by each organism in idiosyncratic ways. There is, for example, a so-called "universal" genetic code by which the nucleotides of genes are translated three at a time into the amino acids of proteins. But many organisms have slightly different codes. Thus, whereas in many areas of science one gathers data to establish a point and, once the point is established, one is done with the data, in biology the data are central and are referred to again and again.

The intermediate goal includes coarser-resolution maps. We are still very far from having the complete sequence. At present only about 6 million nucleotides of human sequence (about 0.2 percent of the total) are known. Furthermore, the cost of determining the sequence is currently too high (on the order of $1 per nucleotide) to contemplate an immediate drive to obtain the full sequence. Fortunately, much useful information can be obtained without sequencing. Maps of lower resolution than the sequence can be based on various sorts of landmarks—features of a chromosome detectable in some experiment. The distances between such landmarks are typically measured in ways that give one a very rough approximation of the number of base pairs between them. All such maps may be considered to be conceptually built on the (yet unknown) sequence as a coordinate system.

One technique with immediate medical application is linkage mapping. Chromosomes break and recombine fairly frequently as the genetic material

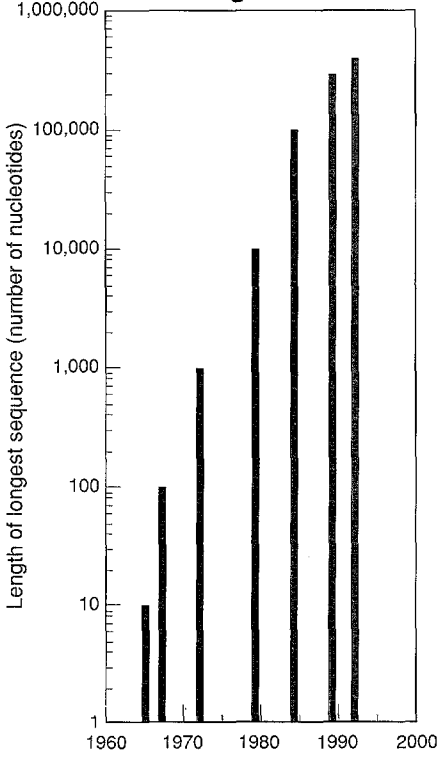
Decades of Nonlinearity: *the growth of DNA sequence data*

Christian Burks, Michael J. Cinkosky, and Paul Gilna

The first nucleotide sequence was published in 1965; it was the sequence of an RNA molecule less than 100 nucleotides long. The methods used were so arduous that until the mid-1970s a person could determine the sequence of only about a hundred bases in a year. Then Maxam and Gilbert in the U.S. and Sanger in England developed new sequencing techniques that were a hundred times faster (see "DNA Sequencing" in "Mapping the Genome"). Figure 1 shows that today biologists are determining the complete sequences of pieces of DNA over 100,000 nucleotides in length. Almost 100,000,000 nucleotides of sequence data have been published—a wealth of information that has formed the basis for many scientific discoveries. How has the enormous and rapidly growing quantity of data been maintained and managed?

As shown in Figure 2a, the rate of sequence-data accumulation was increasing rapidly in the late 1970s. (Data for Figure 2a were compiled from the GenBank database, which includes the publication date and length of each sequence entered.) In response to the growing interest in gathering and analyzing the data, the biology community held several discussions in 1978 on establishing a database facility to collect, organize, and distribute sequence data and annotation about each sequence. For design purposes, the operation of a database can be compared to industrial processes in which a set of input objects is transformed into a set of output objects. In a sequence database, the input is DNA sequences generated by individual laboratories and stored in individual formats with varying amounts of annotation; the output is a collection of DNA sequences stored at a central facility in a uniform format with a precisely defined degree of annotation. For any such process to be workable and efficient, the mechanism for the process must match the volume of the input stream.

Figure 1



During the planning stages for the public sequence databases, how fast did biologists expect the amount of data to grow? Up to 1981 the few recorded projections generally assumed linear growth. Figure 2b shows a linear projection—based on the average annual rate from 1975 to 1977, 25,000 nucleotides per year—for the period up to 1986. (Note that the scale of Figure 2b compresses the previously impressive growth up to 1978.) The linear model predicts that under 300,000 nucleotides of sequence data would have been accumulated by 1986, and that a database project would have had to handle no more than 30,000 in any year. Funding-agency planning and subsequent project proposals to the agencies were based on that linear model. In 1982 the GenBank project, the American sequence database, was established at Los Alamos through a five-year contract with the NIH. (Also in that year a database storing essentially the same information was established at the European Molecular Biology Laboratory; Japan developed a similar institution a few years later.) Because a steady rate of data accumulation was expected, GenBank was staffed with only a few people who were expected to search the literature and enter into a database all the DNA and RNA sequence data that would appear.

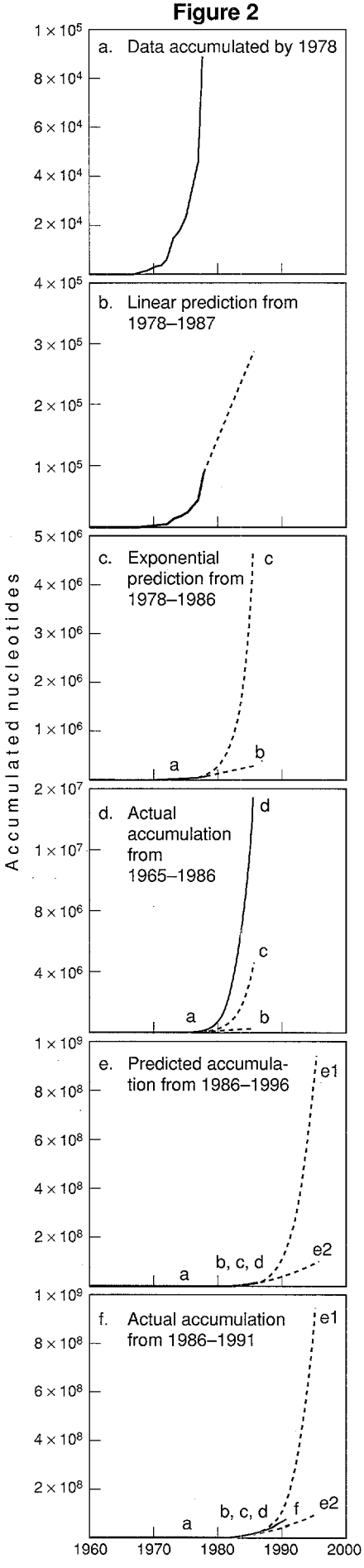
Suppose the community had instead projected exponential growth for the sequence data. Figure 2c shows that if we use the annual rate increase for the years 1975-77 (64 percent per year) to project the accumulation over the period 1978-86, an exponential model predicts an accumulation 15 times that of the model in Figure 2b, and a rate of accumulation orders of magnitude higher. Clearly, in that scenario a database project could not rely on a constant number of staff members each processing data at a constant speed.

What really happened? As can be seen in Figure 2d, the increase of sequence data far outstripped even the exponential model, and completely dwarfed the linear model that was actually used to design GenBank. This created a crisis for the scientific community wanting access to all these data and in particular for the GenBank project, which was responsible for providing access.

In 1986-87, as we planned and developed proposals for the second five-year GenBank contract, we revisited the issue of modeling the growth of sequence data. Figure 2e presents the envelope in which we expected the growth to lie. The lower limit is an extrapolation from the previous three years assuming a constant rate of acceleration. The upper limit is based on the assumption that seven billion bases of sequence, twice the total of the human genome, will be determined by 2005 (consistent with the goals of the Human Genome Project). The rate of acceleration is assumed to increase linearly to bring the curve to that endpoint. With the genome project in mind, we developed a new strategy—and corresponding mechanisms—for the flow of data in and out of the database (see "Electronic Data Publishing in GenBank" below) that we believed would accommodate growth within the projected envelope shown in Figure 2e.

Five years later, Figure 2f shows that actual growth of sequence data has indeed remained within this envelope, and that the accumulation of nucleotide sequence data continues to accelerate. It is worth noting that if the Human Genome Project goals for sequencing are to be met, the rate of sequencing will have to accelerate considerably over the next decade. ■

Figure 2



Further Reading

Walter B. Goad. 1983. GenBank—and its promise for molecular genetics. *Los Alamos Science* 9 (Fall): 52–61.

C. Burks, J. W. Fickett, W. B. Goad, M. Kanehisa, F. Lewitter, W. P. Rindone, C. D. Swindell, and C.-S. Tung. 1985. The GenBank nucleic acid sequence database. *Computer Applications in the Biosciences* 1:225–233.

Christian Burks. 1989. How much sequence data will the data banks be processing in the near future? In *Biomolecular Data: A Resource in Transition*, edited by R. R. Colwell, pp. 17–26. Oxford University Press, England.

Christian Burks. 1989. The flow of nucleotide sequence data into data banks: role and impact of large-scale sequencing projects. In *Computers and DNA*, edited by G. Bell and T. Marr, pp. 35–45. Addison-Wesley, Reading, MA

Michael S. Waterman. 1990. Genomic sequence databases. *Genomics* 6:700–701.

is passed from parent to offspring, and measuring the frequency with which two traits are inherited together allows one to calculate the probability that the responsible genes are on the same chromosome, and if so, about how far apart they are. Linkage mapping has been used successfully to find the approximate location of several disease genes, as a first step in the process of locating and studying the defect. The cystic-fibrosis gene was recently isolated in this way, leading to a much clearer understanding of the disease.

Thus the intermediate goal of the Human Genome Project is an atlas of maps containing one map for each chromosome. Each map is conceptually an annotated sequence, although the sequences are, at the moment, very sparsely filled in.

A complication in this picture is that most groups currently maintain separate maps for linkage-mapping data, sequencing data, and data resulting from other techniques. This is because of disparities in units of measurement. Distances measured in linkage experiments, for example, are expressed in morgans. (The distance in morgans between two sites is the average number of recombination events between them in one meiosis—one set of cell divisions producing an egg or sperm.) But because frequency of recombination at a particular site on the chromosome depends strongly on the (usually unknown) nucleotide sequence at the given site, distances in morgans do not translate by any fixed formula to distances in nucleotides. Nevertheless, we will show below that it is both possible and profitable to integrate these different views of the chromosome into a single map. As well, differences between individuals (there are several billion human genomes, not one) may be best represented as variants within a single comprehensive map.

Both the creation and communication of maps depend on computational tools

Computation plays a central role in almost every facet of the Genome Project. This may come as a surprise, since biology has not traditionally been as heavily computational as, for example, physics or chemistry. But molecular biology is different from traditional biology, and the Genome Project accentuates the differences. There follow two examples.

Disperse workgroups depend on complex communication. Since maps are of perennial interest, and also grow and change daily, there is a great need for instantaneous communication between the producers and consumers of map information. The need for continuous communication is currently most often seen in working groups spread across several laboratories and engaged in the search for a single disease gene. A good example is found in the consortium of laboratories searching for the genetic defect which leads to Huntington's disease.

In such groups continuous communication is often now maintained by faxing text or drawings of maps. However, maps are rapidly growing too complex to manage in this way. In order to track positional information on thousands of map elements at many levels of resolution, undergoing frequent revisions and additions, one needs highly structured databases linked by computer network to graphical interfaces at many sites. This key computational need will require significant development beyond what is currently available.

In the next section we will discuss the major challenges in information management for the Human Genome Project.

Recognition of significant patterns in sequence data depends on sophisticated analysis. Computation also plays a central role in discovering the language of the genome. Many biologically significant patterns in sequence data are invisible to the eye, but can be detected with the aid of computation.

Such insight comes frequently, but an early example is still one of the prettiest. In 1983 R. Doolittle and his colleagues were comparing newly determined sequences to sequences archived in existing databases, and discovered that the transforming (that is, cancer-causing) protein $p28^{sis}$ produced by simian sarcoma virus was remarkably similar to platelet-derived growth factors (PDGFs), proteins whose function in stimulating cell growth was well known. This discovery suggested the natural hypothesis that the sarcoma (connective-tissue cancer) caused by $p28^{sis}$ results from a malfunction in the normal biochemical pathways for PDGFs. Though the cancers are still imperfectly understood, the hypothesis seems to be sound. It has been shown that in the transformation process $p28^{sis}$ interacts with the normal cellular receptors for PDGFs.

In the final section of the article we will discuss the current state of the art in computer interpretation of sequence data.

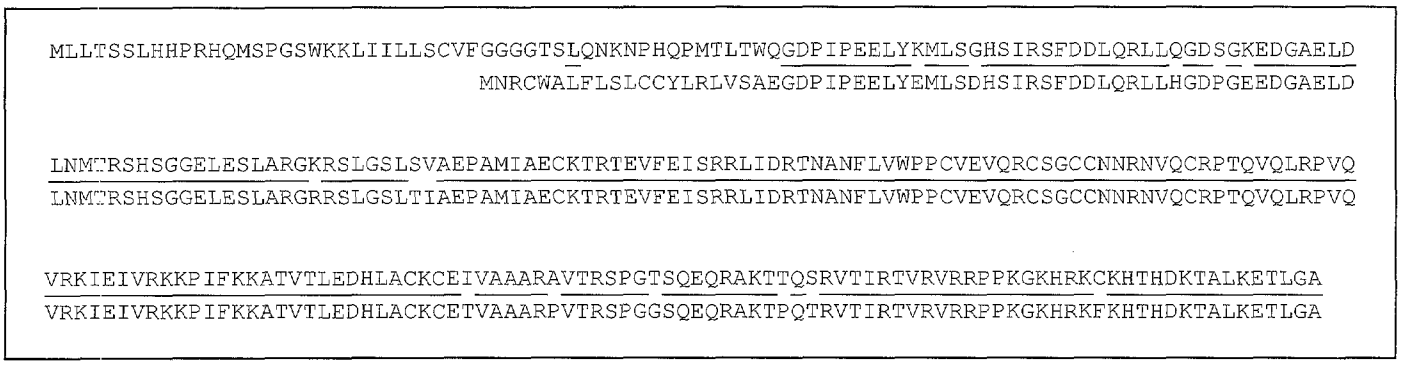


Figure 2. Sequence Alignment between a Sarcoma Oncogene and a PDGF

The upper sequence is the amino-acid sequence of the precursor of the cancer-causing protein p28^{sis} produced by simian sarcoma virus, as translated from nucleotides 3657 to 4772 of the virus's genome. The lower sequence is that of the precursor to a human protein, c-sis/platelet-derived growth factor 2, as translated from cDNA. Lines between the sequences indicate identical amino acids. The conspicuous similarity between the two proteins suggests that the viral gene originated through incorporation into the virus's genome of human sequence or similar sequence from another primate. Moreover, SIS/PDGF2 promotes normal cell growth and its mRNA has been found in tumors, suggesting that p28^{sis} causes cancer by a mechanism related to the functioning of SIS/PDGF2. (The amino-acid abbreviations are A, alanine; C, cysteine; D, aspartic acid; E, glutamic acid; F, phenylalanine; G, glycine; H, histidine; I, isoleucine; K, lysine; L, leucine; M, methionine; N, asparagine; P, proline; Q, glutamine; R, arginine; S, serine; T, threonine; V, valine; W, tryptophan; Y, tyrosine.)

The Human Genome Project requires advances in information management

Many components of an information-management system for the Genome Project already exist—commercial database management systems (DBMSs), computer networks, and hardware for graphical display—but many of the components specific to biology have yet to be developed. For example, though for fiscal accounting systems the data categories and transactions have been standardized for many years, the language in which an emerging description of the genome is being written changes and expands frequently. Without being comprehensive, we present in this section a few of the key problems and how they are being solved.

Efficiency is a natural focus at the stage of covering ground. In the early stages of a mapping project, when a large portion of the map-to-be is "terra incognita," the main business is simply data acquisition, and a key focus of the project engineers is efficiency in the data-acquisition process.

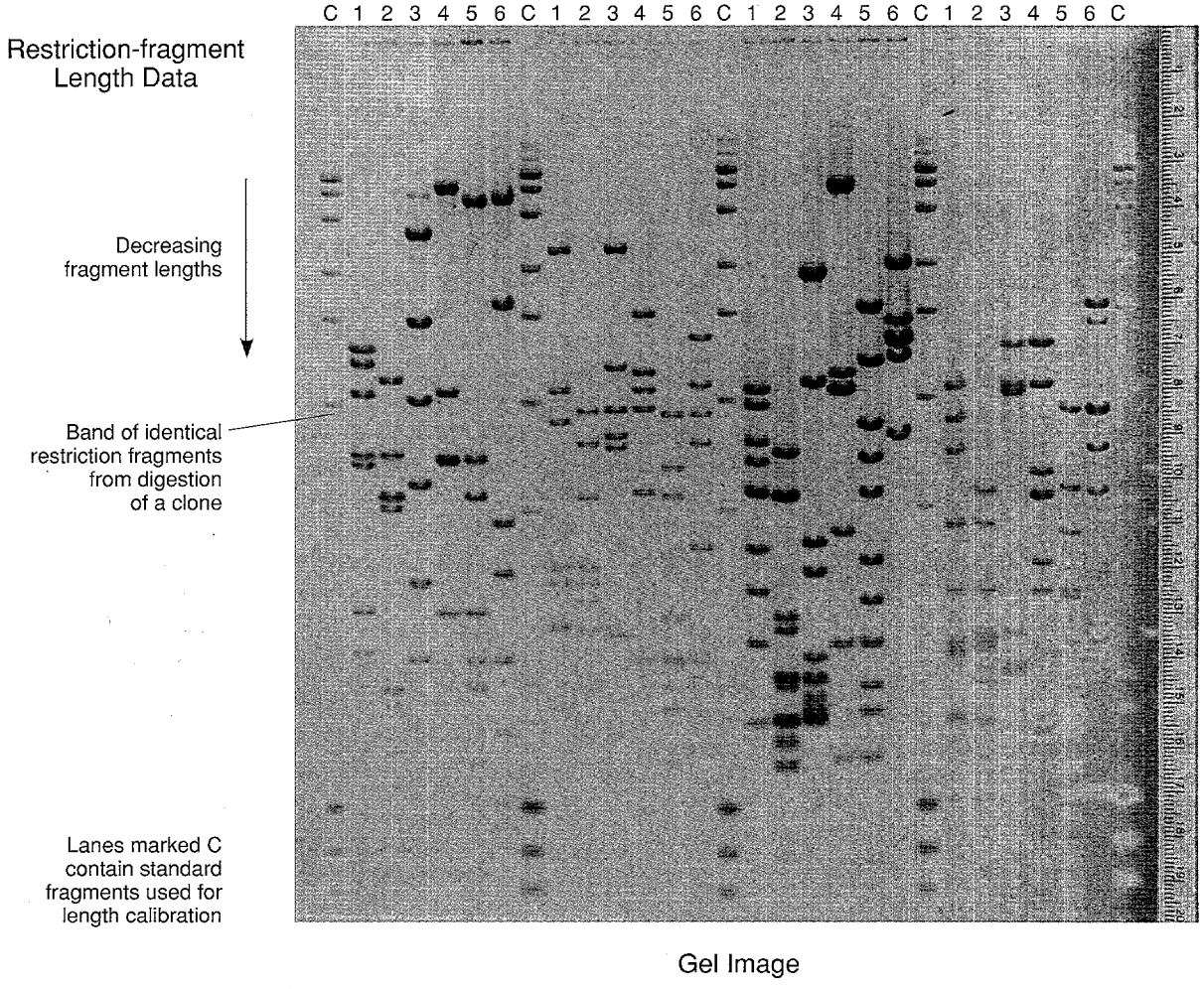
LANL is placing great emphasis on building a "physical map" of human chromosome 16. A physical map is one which gives access to the DNA of any region, and is made by determining pairwise overlaps among a large number (about 4000 at Los Alamos) of cloned segments of DNA, and then deducing the arrangement of the clones relative to each other and to the chromosome (see "The Mapping of Chromosome 16"). It almost goes without saying that an electronic database is required for efficient information processing in a mapping project the size of that at LANL. To give some idea of the complexity of the information we note that the physical-mapping database at Los Alamos currently tracks the sizes and sources of approximately 100,000 fragments of DNA from chromosome 16, and records over 7,000,000 pairwise positional relationships relevant to the emerging map.

The Los Alamos database is currently implemented in the Sybase Relational Database Management System (DBMS) on a network of Sun workstations. Because the Sybase software handles the network transparently, it appears to each project participant as if all the data were stored and immediately available on his or her own desktop.

SCORE: *a program for computer-assisted scoring of Southern blots*

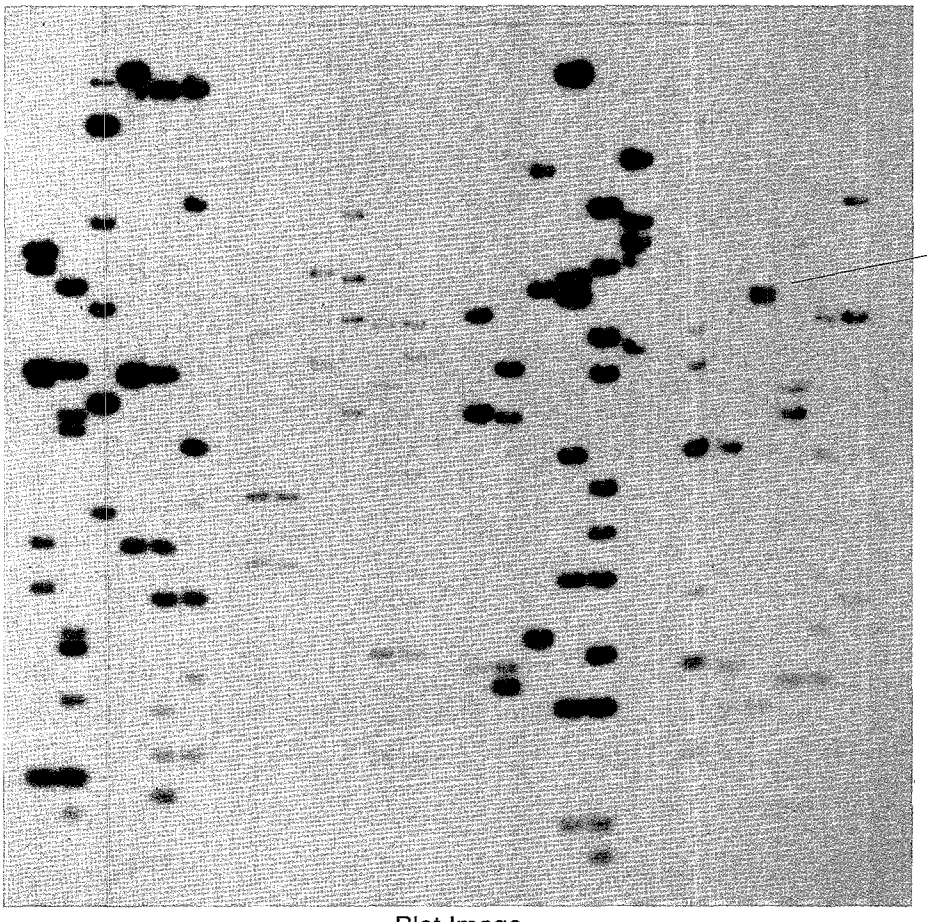
T. Michael Cannon, Rebecca J. Koskela, Christian Burks, Raymond L. Stallings, Amanda A. Ford, Philip E. Hempfner, Henry T. Brown, and James W. Fickett

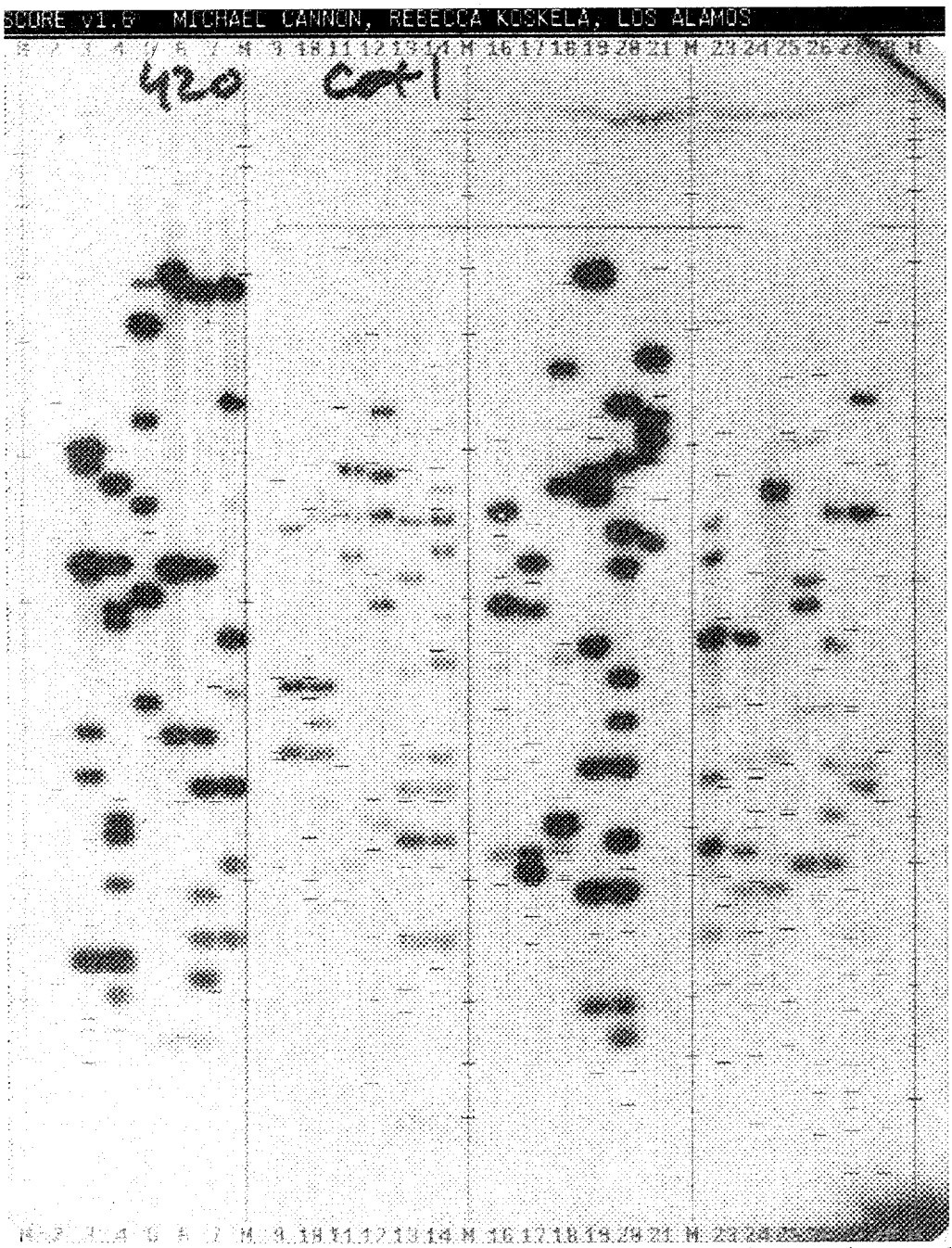
The Human Genome Project aims to collect unprecedented (for molecular biology) amounts of information, so the transfer of repetitive tasks to machines is essential. As part of the LANL physical-mapping effort, we have partially automated the task of entering clone-fingerprint data into computers. One aspect of the automation was the development of a simple image-manipulation program called SCORE. This program has improved the accuracy of the data entry and sped up the process by an order of magnitude.



As explained in "The Mapping of Chromosome 16," the Los Alamos physical-mapping project uses clone fingerprints that consist of two kinds of data. The first is a list of the lengths of DNA fragments obtained by digesting a larger cloned fragment with a restriction enzyme and then separating the restriction fragments by length using gel electrophoresis. On the previous page appears a sample photograph of a gel. The gel is divided into vertical lanes, each lane containing all the fragments of one digest of one clone. Every clone is subjected to three digests, so there are three lanes of fragments from each clone. Each fuzzy horizontal band within a lane consists of identical restriction fragments from the digest contained in the lane. The band's vertical position gives the length of the fragments in it.

The second kind of data is a Southern blot of the gel that indicates whether or not (or to what degree) certain repetitive sequences are present in each restriction fragment. The figure below is a blot image produced by hybridization of repetitive sequences to the gel shown on the left (see "Hybridization Techniques" in "Understanding Inheritance"). Bands of fragments produce a signal on the blot image only if they contain the particular repetitive element being tested for.

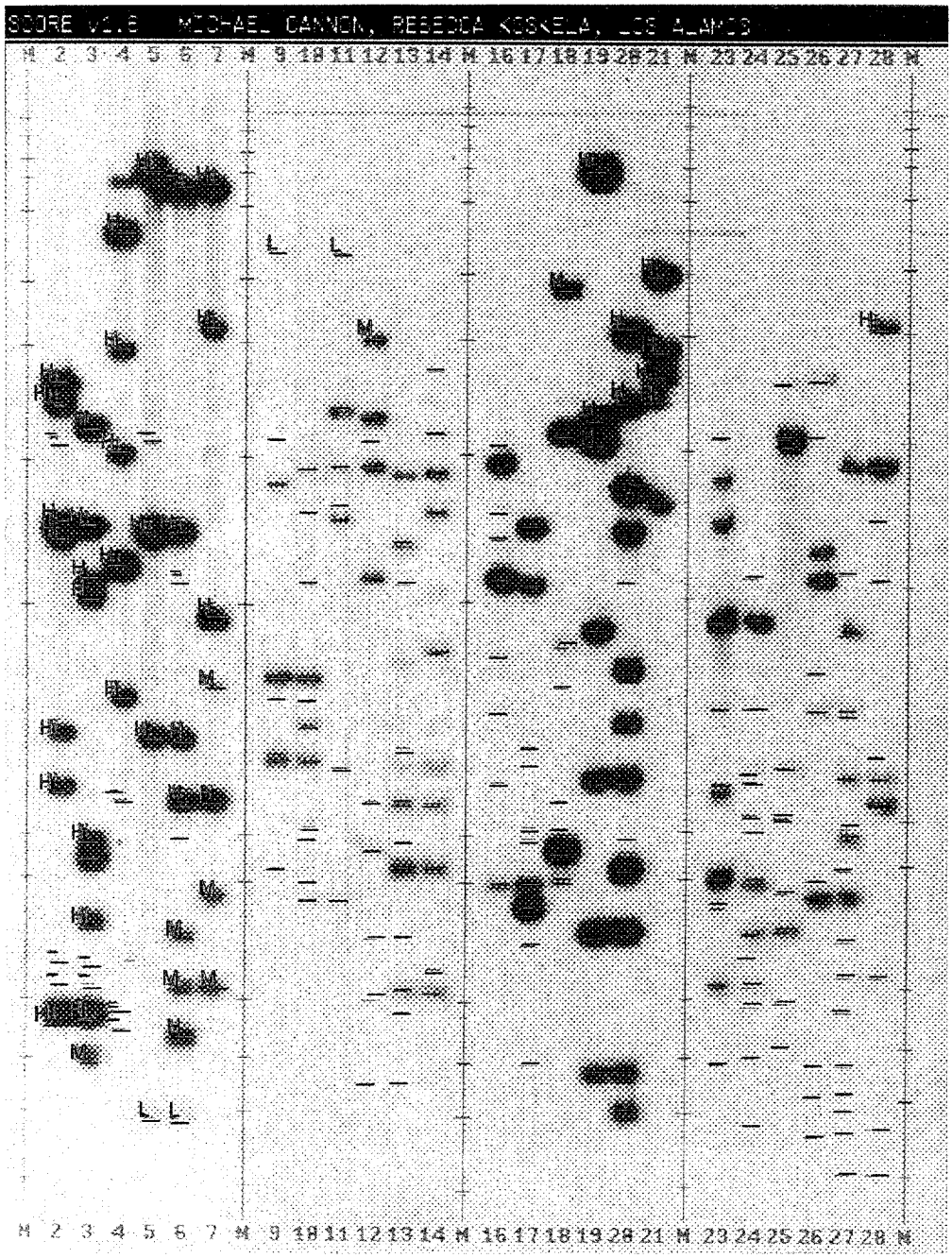




The blot image is used to assign a score to each band indicating the strength of its hybridization signal, a process known as scoring the blot. Therefore the bands on the blot image must be matched with the corresponding bands on the gel image. Formerly the two images were matched by hand, one region at a time. Each fragment was identified manually by numbering the lanes and bands on the photographs. After the scores were assigned, they were typed into our mapping database in a separate operation. Scoring the blot was the most labor-intensive part of fingerprinting. Now we score blots on a scientific workstation using the SCORE program.

Before SCORE is run, the fragment lengths are determined by a commercial image-processing workstation. Another program takes the report from the image processor and stores the lengths in the database. Also, the blot image is digitized using a desktop scanner. SCORE retrieves the fragment lengths from the database and constructs a schematic of the gel image in which the bands are denoted by colored horizontal lines positioned according to their length. The program then superimposes the digitized blot image on the schematic gel image. The figure above shows the two images on the previous pages as stored in the computer and superimposed; they match only approximately.

When the two images are on the screen, the user chooses two points on each image that should be aligned. The program then resizes and moves the digitized blot image to align it with the schematic gel image. The figure at right shows how the user sees the two images overlaid and matched on the computer screen.



At this point the actual scoring takes place. The user points to a band with a mouse, is given a menu of possible scores, and chooses one. Thus the program retains the use of expert human judgement where necessary. SCORE displays the score chosen, next to the band, for the rest of the session (colored letters in the figure). Any score may be revised at any time. If a band shows on the blot image but not on the gel image, the user may add a new fragment to the database. When all fragments have been scored, the program places their scores directly into the database, each score being associated with the proper fragment.

This program has not only cut the time needed for scoring the images by a factor of ten, but it has eliminated typographical errors in data entry. Using SCORE also has the advantage that the complete fingerprint data are in a database, easily accessible by network to the whole group working on the project and readable by the map-construction software, from the moment they are first determined.

For most genome projects, including that at LANL, interface software that translates between internal storage format and the users' intuitive view of the data is developed locally. The accompanying sidebar, "SCORE: a program for computer-assisted scoring of Southern blots," shows one specialized graphical editor which has facilitated rapid and error-free data entry.

Building and maintaining such interface software is itself a formidable task. Efficiency in the software development process is therefore as important as efficiency in the primary task of data acquisition. So although it would be pleasant to have specialized interface software for each data-processing task, there is a need for some more general and less expensive interface. This need is especially acute because experimental techniques and strategies for mapping are constantly changing as biotechnology advances, so that specialized software often has a rather short lifetime.

This need for a general and inexpensive interface has been met by a "database browser" developed by Robert Sutherland at Los Alamos. Someone using the browser sees any of a set of similar screens, one for each type of object in the database. (Types of objects include clone, clone overlap, and DNA sequence.) An example of a data screen is shown in Figure 3. All the screens follow the same style, making the browser easy to learn. Each one lists both the attributes of the current object, and also the other kinds of objects related to the given one. One can retrieve data either by filling in known attributes and asking the software to complete the form, or by following links from one object to related ones. Thus the browser provides access to all data in the database without requiring the user to know a specialized query language.

The current version of the browser is quite easy to maintain, because all the screens are derived from a template set of forms and procedures, in a relatively straightforward way. Nevertheless, every time the database structure changes (a not infrequent occurrence, as experimental methods and strategies change) some custom programming is needed. A new version of the browser is planned, in which the browser software itself will be capable of reading the database structure and configuring itself to match. We think the new version will be invaluable to other laboratories newly setting up mapping efforts, enabling them to put in place a rudimentary data-management system very quickly.

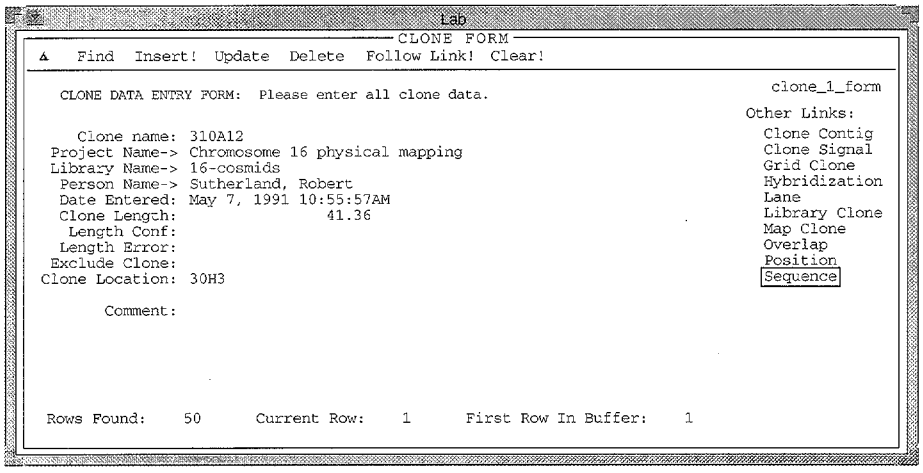


Figure 3. The Clone Screen in the Database Browser

Attributes of the clone include, for example, the name of the project using it and the clone insert length. Related objects include sequences, for example; if the user highlights "sequence" at the right of the form, and then clicks on the button "follow link," any sequences derived from this clone will be retrieved.

Map definition is a natural focus at the stage of mature results. In the fourteenth century, when maps were mostly local, it was possible to make reasonable maps assuming the earth was flat. In the age of exploration, however, the science of map making came to depend on a clearer understanding of the shape of the earth, and on an analysis of the distortion resulting from projecting a spherical surface onto flat paper. Similarly, now that the Genome Project has accumulated mapping data that cover several large regions of the genome fairly densely, it is time to consider carefully just what genome maps are and how we should go about constructing them.

It might seem as if a one-dimensional map of a DNA molecule should be trivial, or at least that it should be simpler than a geographic map. But in fact genome maps are more complex than geographic maps in at least three important ways. Two of these—the use of incommensurable units of distance and the variation among six billion humans—have already been mentioned.

The third is a high level of ambiguity in the data. Given two known points on the earth's surface, it is straightforward to estimate the distance between them. But given two genes or two fragments of cloned DNA, it is typical to go to considerable trouble only to estimate the probability that they are adjacent. Distance relationships are probabilistic not only because the mapping experiments give only partial information,

but because copies of many genes and other sequences occur more than once in the genome with only small differences. Since all physical-mapping methods depend on sequence similarity to determine whether two pieces of DNA are identical, mapping experiments sometimes indicate overlap where there is none. Derivation of a consensus map from fuzzy, probabilistic data is one of the more interesting and important challenges in the Genome Project.

Map construction is an optimization process based on fuzzy objectives. Probably because of the analogy to more familiar geographic maps, investigators often see the map-building process as fundamentally incremental. That is, at any given stage of map construction, one takes as given the map as it stands so far, and looks for the best way to add new data to the existing structure. Even in the apparent exceptions to this practice, as when a committee attempts to reconcile two contradictory maps, one can observe a fundamentally incremental approach—to save as much as possible of an existing structure and add new, or contradictory, data in as conservative a way as possible.

But recognition is growing that map construction requires a global, non-incremental procedure. The reason is simple—as long as the data are probabilistic, it is likely that parts of the map as constructed so far are wrong, so that the entire map needs to be reconsidered when new data come to light. (For example, among those pairs of DNA clones which have a 0.9 probability of overlap, we expect, by definition, one pair out of ten not to overlap.) Therefore one should treat map construction as an optimization problem. Adopting this point of view, one takes all the probabilistic statements about positions as a large set of objectives which a "good" map should fulfill, and attempts to reconcile them all simultaneously, as well as possible, in a consensus map. Calculating an explicit fitness for maps, rather than relying on intuition is, though mathematically routine, a novel idea for many physical-mapping groups. The definition of a good criterion for fitness is a difficult problem; it will probably not be solved in a standardized way for some time.

As input to the optimization problem, it is important to correctly state the objectives. That is, whereas current procedure is often to interpret raw experimental data by placing a new point on the map directly, there should be an intermediate step of recording the results of the experiment alone—an overlap between two clones, say, or a localization of some clone to the region between two known genetic markers—with realistic ambiguity in position and probability.

For the optimization itself, a number of techniques might be applied, including linear programming, simulated annealing, and genetic algorithms. We (the author, M. Cinkosky, and D. Sorensen) have adapted genetic-algorithm techniques to develop an optimization algorithm for assembling physical maps. We chose the genetic-algorithm techniques because the overlap data often contain apparent contradictions and genetic algorithms are known to be robust in the face of such data, and also because the map objectives are not naturally stated as linear equations or inequalities. The input to our algorithm can be clone-overlap data from any kind of experiment, as long as the data fit into the categories of overlap likelihoods, estimated overlap extents, and estimated clone lengths. For computational efficiency, the input clones must be divided into *a priori* contigs in which each clone is connected to the others by a chain of overlaps all having probabilities greater than 0.5. The genetic algorithm then searches for an arrangement of the clones in a contig which fits the experimental data well, but does not try to determine the overall arrangement of the contigs on the chromosome. The algorithm is called GCAA, for Genetic Contig Assembly Algorithm. Figure 4 illustrates GCAA as it is used in LANL's chromosome-16 mapping project.

A genetic algorithm operates by a simulation of evolution. GCAA begins by constructing a population of a few hundred different arrangements of the clones assigned to an *a priori* contig. In each arrangement, called a GCAA-chromosome, every clone is randomly assigned a length close to its measured length. Every clone is also assigned a position to the right of an arbitrary starting point. The analogy to evolution is that GCAA-chromosomes "mate" and produce "children" whose characteristics are determined by a process resembling genetic recombination. Then only the "fittest" GCAA-chromosomes survive to mate in future generations.

GCAA calculates the fitness of each GCAA-chromosome by checking how well it corresponds to the data, with discrepancies from the most certain data points given the most weight. Three separate measures of fitness are computed: one for the overlap probabilities, one for the overlap lengths, and one for the clone lengths. For the overlap-likelihood and clone-length data, discrepancies from the most certain data points are given the most weight.

In the core of the algorithm, the following procedure is carried out repeatedly: GCAA selects a "tournament" of four GCAA-chromosomes at random. The two chromosomes whose clones have the most disparate positions then "mate" and produce two "children." In each child of the mating, some of the clones are positioned as in one parent, and the other clones have their arrangement taken from the other parent.