PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT.

Toward a Human Genome Encyclopedia

Kaoru Yoshida [1,1], Cassandra Smith [2],
Toni Kazic [3], George Michaels [4], Ron Taylor [4],
David Zawada [5], Ray Hagstrom [6], Ross Overbeek [6]

[1] Division of Cell and Molecular Biology, Lawrence Berkeley Laboratory, Berkeley, CA 94720, U.S.A.
[2] Department of Molecular and Cell Biology, University of California at Berkeley, Berkeley, CA 94720, U.S.A.
[3] Department of Genetics, Washington University School of Medicine, St. Louis, MO 63110, U.S.A.
[4] Division of Computer Research and Technolgy, National Institutes of Health, Bethesda, MD 20894, U.S.A.
[5] Advanced Computer Applications Center, Argonne National Laboratory, Argonne, IL 60439-4832, U.S.A.
[6] Division of Mathematics and Computer Science, Argonne National Laboratory, Argonne, IL 60439-4832, U.S.A.

Abstract

Aiming at building a human genome encyclopedia, a human genome mapping database system, *Lucy*, is being developed. Taking chromosome 21 as the first testbed, more than forty maps of different kinds have been extracted from publications, and several public and local genome databases have been integrated into the system. To our knowledge, Lucy is one of the first systems that have ever succeeded in genome database integration. The success owes to the following key design strategies: (1) A sequential logic programming language, Prolog, has been used so that the database construction and query management could rely on the internal database facility of Prolog. (2) An object-oriented data representation has been employed, so that any kind of data could be manipulated in the same manner. (3) A mini language, *map expression*, has been designed, which enables map representation in a relative-addressing manner and also linkage of one map to another. These strategies are applicable for building a genome mapping database not only on human chromosome 21 but also beyond chromosomes and beyond species.

1 Introduction

1.1 Why Biological Applications?

The fact that only four DNA bases (adenine, thymidine, guanine, and cytosine – symbolically represented as A, T, C and G respectively) encode most of the information on current life and its history is fascinating from the viewpoint of computer science. More interesting is that many biological reactions are due to the property that A and T make a complementary pair as well as G and C do. Genome analysis is potentially a large application area for symbolic computation. As biological experimental methodology develops, more gene information is accumulated and analysed. This holds especially true for such large scale models as the human genome whose total genome size reaches a few billion of bases. Since NIH (National Institute of Health) and DOE (U.S. Department of Energy) embarked a joint national research initiative [30, 31], human genome projects have been initiated in many other countries and research activities are being expanded and accelerated day by day [89, 65, 83]. To proceed efficiently in the ever accelarating climate of current biological research, strong support and feedback from computer-aided analysis is mandatory [74, 53, 39].

1.2 What is a Physical Mapping Process?

Genome mapping is similar to geographical mapping. The genome mapping is now akin to the early times of geography. First of all, it is not known yet exactly how big the genome is. Continents, countries, states, cities and streets work as *geographical markers* which give positional information, *addresses*, on the earth. As well, continental-level landmarks with location-specific information such as a single copy DNA sequence (i.e., sequences that occur only once in the genome) [70] have been discovered here and there on the genome; fragmentary maps around these landmarks are being drawn, some of which are being glued one to another. Furthermore, as there are geographical maps and time-zone maps, there are different kinds of genome maps, roughly categorized into two kinds: physical maps giving physical distances (i.e., the number of bases lying) between the markers and genetic maps giving recombination frequencies between the markers. This section introduces what is genome physical mapping. It should help in understanding the genomic data which will be involved in the genome mapping databases described in later sections. For more details, consult [90].

Chop, Identify and Assemble. Figure 1 shows how physical mapping is done. In general, a genome is too large to be directly sequenced[2] with the current sequencing technology. For example, the total size of hu-

[2] *DNA sequencing* means experimentally reading a DNA sequence consisting of A, T, C, G bases.

man chromosome 21, the shortest human chromosome, is thought to be 50 to 65 Mb (mega bases), while the maximum length of DNA read per day is about 500 bases, including reading error corrections, and the cost of sequencing is about one dollar per base [44, 9]. If the chromosome 21 were read in a serial manner, it would take 250 years. Hence, first of all, the chromosome has to be excized into pieces (called *fragments*), which are small enough for further analysis, such as a 20-30 Kb (kilo bases) to a 2-3 Mb (*step (1)*). The excision is done by a physical method (e.g. by irradiation [27]) or by a chemical method (e.g. by digestion with restriction enzymes[3]) (*step (3)*). Then, the DNA fragments are to be assembled to the original chromosome. For the assembling, there are a variety of methods, depending on (a) whether or not the DNA fragments may overlap, (b) how the overlap and adjacency of the DNA fragments are detected, and (c) whether each step of experiment is attempted against an individual fragment or to a group of fragments at a time.

A Conventional Physical Mapping Method. Figure 1 shows a conventional mapping method which deals with non-overlapping restriction fragments. This method starts with chemical digestion using a restriction enzyme. The restriction fragments are sorted in size (by the electrophoresis method (*step (4)*) and assigned to an approximate region through hybridization[4]. Each fragment is hybridized to a variety of cell lines[5]. As each cell line covers a different region, the pattern of hybridization signals against different cell lines determines which region the target fragment resides (*step (2)*). Next, hybridization is attempted against probes within that region. With a positive hybridization signal on with a probe, the fragment is determined to lie *around* the address of the probe (*step (5)*).

A clone containing a specific restriction site is called a linking clone. A linking clone is split at the restriction site and then each half is hybridized to complete digests[6]. As the two halves are known to be next to each other, complete digests fished by the halves of a linking clone are found to be adjacent. Thus, linking clones introduce the notion of adjacency that works as a strict constraint in linear-ordering restriction fragments.

As a result of hybridization against a number of probes, fragments are eventually given a linear order (*step (6)*). The process (3) thru (6) is repeated until a map with the desired precision is obtained.

A New Physical Mapping Method. A new method, called *clone contig assembly*, is shown in Figure 2. This method uses clones of overlapping fragments of almost the same size determined by the cloning vector. By determining overlapping pairs of clones, walking is attempted from one clone to another. The resulting walking path forms an island of contiguous clones, that is called a *contig*. This method has variations depending on how the overlaps are detected (e.g., whether based on the restriction digest pattern of each clone or based on hybridization signals [54]). Furthermore, the overlap detection can be attempted against a group of clones at a time, in common. The feasibility of extracting the maximum amount of information in every step of biological experiment and the potential for automation are attracting much attention to these contig assembly methods [29]. In addition, given a set of overlapping clones, the variation of length and overlaps of clones gives a statistical limit on the number of independent islands which can be constructed from the clones [69, 26, 52]. It should be noted that this method can be carried out, vigorously relying on statistical and computational analysis [14, 37].

In summary, the physical mapping process consists of the three steps: (1) excising the whole DNA into pieces, (2) characterizing every piece through hybridization or digestion, and (3) assembling the pieces. While steps (1) and (2) are done through biological experiments, step (3) is a probablistic combinatorial problem. In order to solve this problem, information retrieval from a variety of genome databases is required together with powerful computational tools.

1.3 Mapping Data and Mapping Knowledge

Section 1.2 introduced the physical mapping process from the viewpoint of biological experiments. The resulting data are published in the form of inventories as shown in Tables 1 and 2.

Identification and Adjacency. Table 1 gives a relation between hybridization probes and restriction fragments obtained by digesting cell line WAV-17 with restriction enzyme *NotI* [88]. For instance, row 1 implies that clone 231C which is a representative of locus D21S3 hybridizes to a 2200Kb complete digest and to two partial digests: a 2200Kb fragment and a 2600Kb fragment.

Section 1.2 introduced linking clones with the notion of adjacency. HMG14l and HMG14s are the two halves

[3] *Restriction enzymes* recognize some specific DNA pattern of four to a dozen of bases and cut a double-stranded DNA at some specific position in the pattern.

[4] A double stranded DNA is formed if each strand contains a complementary sequence to the other. Hybridization is an attempt to make a double-stranded DNA or an RNA-DNA hybrid using this property. By labelling a probe (i.e. the counterpart) with an isotope or a dye, by means of autoradiograph or florescence one can detect if the probe has hybridized to the target or not.

[5] *Cell lines* are DNA segments which are generated by deleting a portion of chromosomes or by translocating between different chromosomes.

[6] *Complete digests* are restriction fragments obtained when the restriction enzymes react to completion, i.e., everyone of the target sites is cut. In contrast, *partial digests* are those which contain some fraction of the target sites uncut.

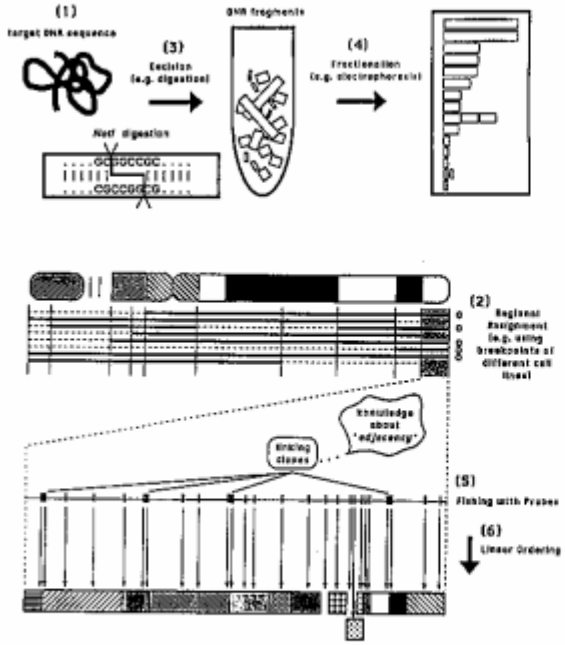

Figure 1: A process of restriction map construction

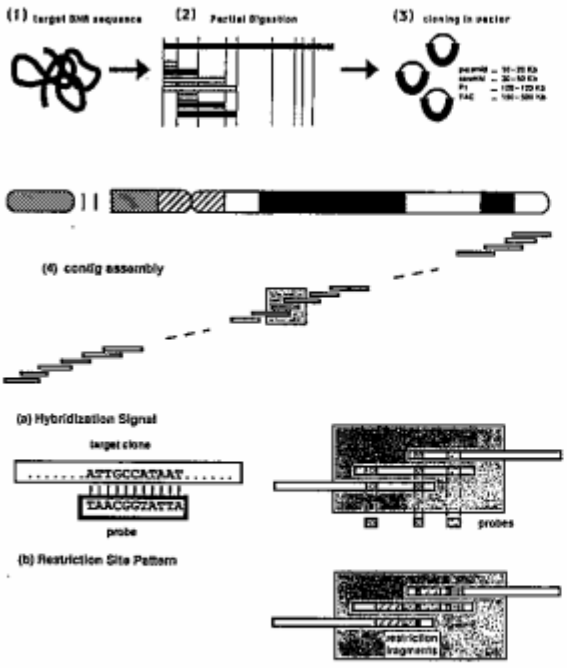

Figure 2: A process of contig assembly

Table 1: Restriction fragments and hybridization probes

	Probe		*Not*I restriction fragments
	locus/gene	clone	
1	D21S3	231C	2200,2600
2	HMG14	HMG14l	75
3	HMG14	HMG14s	300,360,560,630
4		6-40-3	300,360,530,1000
5		D13s	300,2100,2900
6		D13l	75,1800
7	*D21S101	JG373	1800,2100,2300,2600
8	*D21S15	pGSE8	2000,2400,2700
9		LA171l	1800
10		LA171s	750,2100,2350
11	*D21S51	SF93	750,1200,1800,2050,2300
12	*D21S53	512-16P	750
13	*D21S39	SF13A	750
	:	:	

of linking clone HMG14. Hence, the 75Kb fragment hybridized to HMG14l must be next to the 300Kb fragment hybridized to HMG14s. Similarly, for a pair of D13s and D13l, the 300Kb fragment and the 75Kb fragment must be adjacent; for another pair of LA171l and LA171s, the 1800Kb fragment and the 800Kb fragment must be adjacent. The 300Kb fragments in rows 3 to 5 can be interpreted to be identical, assuming that the 360Kb fragment (in rows 3 and 4) be a partial digest containing the 300Kb and the 75Kb fragments and also assuming that the 2100Kb fragment (in rows 4 and 5) be a partial digest containing the 300Kb and 1800Kb fragments.

Thus, given a relationship of restriction fragments and hybridization probes, each restriction fragment is identified using strict constraints such as linking clones and also using its neighborhood information such as a pattern of partial digests.

Confirming Information. In Table 1, the 750Kb fragments in rows 10 to 13 seem to be identical. Also, the ordering of loci D21S101 (row 7) and D21S15 (row 8) is not evident in this table, nor the ordering among loci D21S51, D21S53 and D21S39.

Table 2 shows a relationship of multiple kinds of restriction fragments (of a different cell line, CHG3) and hybridization probes [17] around the same region as Table 1. With an assumption that the *Not*I restriction sites be rather conserved in different cell lines and considering of 10-20% errors in size, the 750Kb fragment in Table 1 can be interpreted to correspond to the 700Kb fragment in rows 4 to 6 in Table 2. The identification of the 750Kb fragments in Table 1 is confirmed by the same set of *Mlu*I digests (<200Kb, 1250Kb and 1400Kb) and *Nru*I digests (600Kb and 2000Kb) found around the 700Kb fragment in Table 2. As for the ordering of loci D21S101 and D21S15, the 1600Kb *Mlu*I fragment in rows 1 and 2 connects D21S101 with D21S3 and the 1400Kb *Mlu*I

Table 2: Multiple digests

	Probe		Restriction fragments		
	locus/gene	clone	*Not*I	*Mlu*I	*Nru*I
1	D21S3	pPW231	700,1800	700,1600	600
2	D21S101	JG373	1400	1000,1600	1400,2000
3	D21S15	E8	1400	1250,1400	1400,2000
4	D21S39	SF13A	700	<200,1250,1400	600,2000
5	MxA/B		700	<200,1250,1400	600,2000
6	D21S51	SF93	700	<200,700,1400	500
:	:	:	:	:	:

fragment in rows 3 to 6 connects D21S15 with D21S39.

In general, mapping data contain a non-trivial imprecision which clouds their interpretation. Interpretation for a set of mapping data becomes less ambiguous with additional information. It is obviously efficacious to accumulate data until a convincing interpretation is acquired.

1.4 Genome Mapping Databases

Public Databases and Laboratory Notebooks. The constantly growing population of genome databases [80] contains precious few mapping databases even considering different species, such as mouse [47], *Caenorhabditis elegance* [32] and *Eschericia coli* [6]. As for human, GDB (Genome Data Base) [40] is the only public mapping database. It contains information about genes, loci (landmarks), clones, contacts and maps. As for maps, consensus maps are collected each of which contains merely the consensus order of loci, without information on physical/genetic distance between loci yet [75].

Laboratory data that are primary or secondary level of experimental data including image films will someday be available in so-called *laboratory notebook databases* which are now under development [54, 68, 4, 58]. Especially for the contig assembly mapping method for which a computer analysis environment is essential, system development efforts are intensive and have been applied for mapping chromosomes X, 16 and 19 [54, 23, 66].

What seems to be missing in genome databases is a continuous link between public databases and laboratory notebook databases. There is a strong need to compare laboratory mapping efforts against those reported in publications and public databases.

Implementations, Interfaces and Integrations. In terms of implementation strategies, most genome databases, including the above, have been implemented using relational database management systems which are based on a *normal form* (or *flat*) relational model [24]. Also these databases provide a query language (usually SQL) interface for programmers and an interactive window interface for end users, both of which rather directly reflect the underlying implementation. Programmers and users must be knowledgable about implementation issues, such as how each relational table is linked to others.

A high level interface is also required for easily sharing and exchanging data between different databases. Among leading database integration efforts, GenInfo [71, 60] is notable. Three databases: Genbank (DNA sequence database), PIR (protein sequence database) and MEDLINE (medical/biological literature database) are converted into the form of an object-oriented data representation language, ASN.1[7] [72], so that data can be easily exchanged among the databases. ASN.1 has also been applied to the construction of a metabolic compound database [49].

In summary, various kinds of information are involved in the genome mapping process. The integration of different databases is a key issue in proceeding further biological research.

1.5 Goal and Strategies

Many queries issued in the physical mapping process are imprecise, e.g., "Get all information around this locus", and "What are the consensus and differences around this locus in all collected maps?" To address these queries, all related information must first be collected from publications and various databases into a map in which all available information is woven at every location of human genome, i.e. a human genome encyclopedia. Then, using this encyclopedic map, a genomic grammar [81] will interface to the user.

The construction of a human genome database system, *Lucy*[8], has started. Taking chromosome 21 as the first testbed, more than forty maps of different types have been collected from publications, and several public and local databases have been integrated into the system. Currently, the system is ready to answer rather general queries such as shown above. To our knowledge, this is the first integrated physical mapping database that has ever been implemented.

The key design features which have enabled the prototyping of Lucy are:

- logic programming,
- object-oriented data representation and query interface,
- map representation language.

The following sections will describe each of these features in detail.

[7] Abstract Syntax Notation 1, ISO 8824.

[8] The name is derived from the nickname given to the first fossile of hominid [48]. The motto herein is *"For any question on human, ask Lucy"*.

2 Representation of Genome Information

2.1 Exploitation of Logic Programming Featuers

Lucy has been implemented in a sequencial logic programming language, Prolog, for its following features:

1. **Database Facility and Inference Mechanism:** Its internal database facility and inference mechanism enable validation of biological data and rules as knowledge immediately when they are expressed as Prolog predicates (programs). Even if they were expressed as Prolog terms (data) as second order predicates, the inference mechanism could be implemented rather easily in Prolog.

2. **Declarative Expresssion and Set Operations:** Its declarative expression and (built-in) search utilities (e.g., built-in set operations such as `setof` and `bagof`) minimize the amount of programming effort for knowledge representation and database retrieval.

3. **Recursive Queries:** Its capability of handling recursive programming and recursive data structures enables a straightfoward implementation of recursive queries that are hard to be implemented with normal form relational databases and conventional query languages such as SQL [28].

4. **Foreign Language Interface:** It is necessary to have a foreign language interface (which is provided in several Prolog implementations) to other conventional but efficient languages, such as C and Fortran, in order to import and develop the computationally intensive sequence analysis and statistical mapping tools.

5. **Portability:** Lucy should be developed as a real system to be used for biological analysis. The stability and portability of the system are the first priority.

2.2 Object-Oriented Representation and Interface

The hybridization results shown in Tables 1 and 2 in Section 1.3 could be represented as Prolog facts of a flat relational form as follows:

```
%-------------------------------------------------------------------
% table1(Locus,    Clone,    NotIdigests).
%-------------------------------------------------------------------
  table1('D21S3',  '231C',   [2200,2600]).
  table1('EMG14',  'EMG14l', [75]).
  table1('EMG14',  'EMG14s', [300,360,560,630]).
```

```
%-------------------------------------------------------------------
% table2(Locus,     Clone,     NotIdigests, MluIdigests, NruIdigests).
%-------------------------------------------------------------------
  table2('D21S3',   'pPW231', [700,1800],  [700,1600],  [600]).
  table2('D21S101', 'JG373',  [1400],      [1000,1600], [1400,2000]).
  table2('D21S15',  'E8',     [1400],      [1250,1400], [1400,2000]).
                :
```

Then, for every element involved in these tables, such as loci, clones and enzymes, information collected from publications and public databases was stored similarly in a flat relational form. Obviously, as the number and variety of relations increased, it will be accordingly difficult to program and maintain the database in this format, and to remember the exact form of each relation.

Another burden handling various different tables becomes obvious when encoding mapping rules. For example, the following program defines the notion of adjacency introduced with linking clones, namely that two restriction fragments are adjacent if one fragment is hybridized to one half linking clone and the other fragment to the other half linking clone and if the restriction fragments are both complete digests:

```
is_adjacent_to(FragmentA, FragmentB) :-
    is_half_linking_clone(HalfLinkingCloneP, LinkingClone),
    is_half_linking_clone(HalfLinkingCloneQ, LinkingClone),
    HalfLinkingCloneP \= HalfLinkingCloneQ,
    is_hybridized_to(HalfLinkingCloneP, FragmentA, Enzyme),
    is_hybridized_to(HalfLinkingCloneQ, FragmentB, Enzyme),
    FragmentA \= FragmentB,
    is_complete_digest(FragmentA),
    is_complete_digest(FragmentB).
```

Here troublesome is that if hybridization results were stored in various forms, predicate `is_hybridized_to/3` would have to be defined for each kind of digests in each different table, as follows:

```
is_hybridized_to(Probe, Fragment, 'NotI') :-
    table1(_, Probe, NotIFragments),
    member(Fragment, NotIFragments).

is_hybridized_to(Probe, Fragment, 'NotI') :-
    table2(_, Probe, NotIFragments, _, _),
    member(Fragment, NotIFragments).

is_hybridized_to(Probe, Fragment, 'MluI') :-
    table2(_, Probe, _, MluIFragments, _),
    member(Fragment, MluIFragments).

is_hybridized_to(Probe, Fragment, 'NruI') :-
    table2(_, Probe, _, _, NruIFragments),
    member(Fragment, NruIFragments).
```

where `member(X, Y)` is a built-in predicate which succeeds if `X` is a member of `Y`.

To relieve these difficulties, an object-oriented data representation has been adopted in Lucy. The hybridization relationship between a fragment and probes has been embedded as an attribute of the fragment.

2.2.1 Principle

First of all, we recognize that any kind of datum is an object composed of attributes and represented as a Prolog fact, `object/2`, consisting of a functor, `object`, and two arguments, as follows:

```
object(ObjId, Attributes).
```

where

- `ObjId` is an object identifier which is unique in the entire system and is formed of a *class* and a *local identifier* unique within the class;
- `Attributes` is a set of attributes which constitute the object. The internal representation of attributes is encapsulated in the variable, `Attributes`.

Next, we construct general interface methods which allow retrieval of information from an object without knowing how that object is internally represented as follows:

- `class(ObjId, Class)` returns the class of the object.
- `id(ObjId, LocalId)` returns the local identifier of the object.
- `attribute(ObjId, Attribute)` returns an attribute composed of an attribute name and an attribute value.

2.2.2 Examples

Starting with a restriction fragment, let us consider several objects related with this fragment and see what kinds of information are associated with them. Note that, in this paper, the attributes are represented in the form of a list merely for ease of explanation; a different data structure, more efficient in space and access, is used in the real implementation.

1. **Restriction Fragment:** This defines the 750Kb *NotI* fragment appearing in rows 10 to 13 in Table 1, that has been digested from cell line `WAV-17` with restriction enzyme *NotI*. This fragment was hybridized to four probes: `LA171s`, `SF93`, `512-16P` and `SF13A`. This information was obtaipined in an experiment done by `Denan Wang`, April 1991, and appears in a literature, `Saito et al (1991)`.

```
object('LUCY:fragment'('Denan1991:WAV-17/NotI/750#2'),
       [input_date(1991/4/24),
        digested_from(cell_line('WAV-17')),
        digested_by(restriction_enzyme('NotI')),
        probes([half_linking('LA171s'),clone('SF93'),
                clone('512-16P'),clone('SF13a')]),
        size('Kb'(750)),
        source(ref('Denan Wang (April 1991)')),
        references([ref('Saito et al (1991)')])
       ]).
```

2. **Probe:** One of the probes, `SF93`, was offerred by `Cox`, and registerred in a local clone logbook, Plasmid Book, with a local name, `CLS3048`. It has an *EcoRI* site at one end and a *SalI* site at the other end and is cloned in a `pUC18` vector, and is resistant to `ampicillin`.

```
object('PB:clone'('SF93'),
       [input_date(1991/8/8),
        symbol('SF93'),
        information_source(db('PB/ver.89-11-8')),
        if_confirmed(yes),
        lab_number('CLS:clone'('CLS3048')),
        within([locus('D21S51'),region('21q22')]),
        size('Kb'(2.1)),
        clone_sites([restriction_site('EcoRI'),
                     restriction_site('SalI')]),
        vector(vector('pUC18')),
        vector_size('Kb'(2.7)),
        antibiotic(amp),
        source('PB:contact'('Cox'))
       ]).
```

3. **Locus/Gene:** Clone `SF93` is a representative of locus `D21S51` whose information is found in public database GDB.

```
object('GDB:locus'('D21S51'),
       [input_date(1991/7/5),
        information_source(db('GDB/ver.1.0')),
        sources(['GDB:source'('Korenberg et al (1987)'),
                 'GDB:source'('Burmeister et al (1990)')]),
        probes(['GDB:probe'('SF-93')]),
        symbol('D21S51'),
        full_name('DNA Segment, single copy probe SF-93'),
        within([region('21q22.3')]),
        locus_type('DNA'),
        if_cloned(yes),
        assignment_modes(['GDB:assignment_mode'('N'),
                          'GDB:assignment_mode'('S')]),
        certainty(confirmed),
        report(include),
        create_date('Apr 17 1990  1:20:46:000AM'),
        modify_date('Nov 25 1990  2:01:29:460PM'),
        approved_date('Sep  8 1990 11:06:13:320PM')]).
```

4. **Contact:** The person simply referred to as `Cox` in the Plasmid Book is `David R. Cox` whose detailed information is found also in public database GDB.

```
object('PB:contact'('Cox'), Attributes) :-
     object(contact('David R. Cox'), Attributes).

object('GDB:contact'('David R. Cox'),
      [input_date(1991/7/5),
       information_source(db('GDB/ver.1.0')),
       'GDB:idx'('GDB:contact'(1148)),
       symbol('David R. Cox'),
       contact_address(['Univ. of California at San Francisco',
                        'Dept. of Pediatrics/Psych/Biochem',
                        '505 Parnassus Ave., Box 0106']),
       city_address('San Francisco'),
       state_address('CA'),
       post_code('94143'),
       country_name('USA'),
       email_address('rjb@canctr.mc.duke.edu'),
       phone_number('1-(415) 476-4212'),
       'FAX_number'('1-(415) 476-9843')
      ]).
```

5. **Literature:** The mapping effort concerning the above restriction fragment and clones was presented in the literature, `Saito et al (1991)`, as follows:

```
object('LUCY:reference'('Saito et al (1991)'),
       [input_date(1991/4/24),
        kind(paper),
        authors(['Akihiko Saito', 'Jose P. Abado',
                 'Denan Wang', 'Minao Ohki',
                 'Charles R. Cantor', 'Cassandra L. Smith']),
        title('Construction and Characterization of a NotI
               Linking Library of Human Chromosome 21'),
        journal('Genomics'),
        volume(10),
        year(1991)
       ]).
```

Thus, not only biological data but also personal information and literature references are all represented in an object-oriented manner.

2.2.3 Restricting Classifications

Many biological terms have been introduced so far, such as chromosome, locus, gene, probe, clone and restriction fragment, but each of them represents *just a piece of DNA*. For example, when a restriction fragment is cloned, it is called a clone. When it is used for hybridization and gives information as a landmark, it is called a probe. The more biological experiments are applied to an object, the more names and attributes are given to it. Also a set of constraints over attributes forms a new category (or class). For example, when a clone is sequenced and found to contain some restriction site in it, it is called a linking clone; if the restriction site is an *Not*I, then it is called an *Not*I linking clone.

```
object(linking_clone(Id), Attributes) :-
      object(clone(Id), Attributes),
      find_attribute(Attributes, categories(Categories)),
      is_member(linking_clone, Categories).

object('NotI_linking_clone'(Id), Attributes) :-
      object(linking(Id), Attributes),
      find_attribute(Attributes,
                     linking_site(restriction_enzyme('NotI'))).
```

As a principle, objects may have no class when they are created. Classification is made as more attributes are accumulated and properties are found through later experiements.

2.2.4 Broading Classifications

The information sources constituting Lucy cover more than forty maps collected from publications and several different kinds of public and local biological databases, as shown in Figure 3.

In general, in integrating databases, mainly two kinds of strategies are considered: (1) one is to distill the source databases and unite them into a single database, and (2) the other is to preserve the original form of the source database and provide a bridging interface over them.

Similar biological experiments are being done in parallel at different places. As a result, similar data are accumulated in different databases or even in a single database independently. Also, each datum stored in a version of a database might be corrected or changed through later experiments and reported in a later version of the database. In integrating a genome database, preserving the redundancy and inconsistency of data is a substantial effort.

As a result, the second integration strategy taken in Lucy keeps track of the redundancy and inconsistency. The following program provides a bridging interface to bundle clones which are stored in various sources. Any clone can be referred to with a class name, `clone`, while

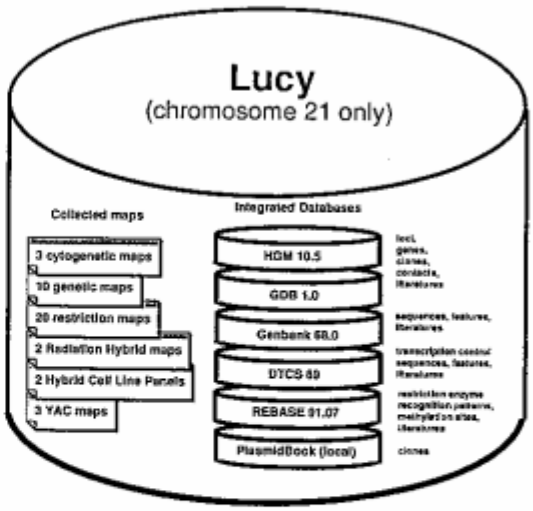

Figure 3: Information sources on chromosome 21, integrated in Lucy

their original class name is preceeded with the database name, such as `PB:clone`. It preserves its origin as an additional attribute, `self(Obj)`

```
object(clone(Id), Attributes) :-
      member(Class, ['GDB:probe', 'PB:clone', 'CLS:clone',
                     'LA:clone', 'LL:clone', 'YZ:clone',
                     'Sakaki:clone', 'LUCY:clone']),
      object_id(Obj, Class, Id),
      object(Obj, Attributes0),
      add_attribute(Attributes0, self(Obj), Attributes).
      ).
```

In summary, the notion of class introduced in Lucy is *loose* unlike such a stringent notion as "class-as-template" which is widely adopted in object-oriented programming languages [41, 78, 91].

3 Constructing a Global Map from Fragmentary Maps

In order to understand mapping information in a visual form, a general graphic interface, GenoGraphics [95, 43], has been hooked up to the Lucy database system.

As shown in Figure 3, those maps collected in Lucy have a variety of range and scaling unit. Some maps cover q-telomeric regions, some do centromeric regions, and many others do some specific region (or island) such as locus D21S13 that is concerned with the Alzheimer disease. Also physical maps are measured their coordinates in Kb (kilo base), genetic maps are in cM (centi-Morgans), and cytognetic maps are in ratio (%).

For the moment, even the total genome size of chromosome 21 is not precisely determined. If every object in maps measured in percentage were specified with an

absolute coordinate, the coordinate would have to be modified every time the total genome size is corrected through later experiments. Similarly, the exact position of the D21S13 locus is not fixed, either. Every time a more precise position were determined for locus D21S13, the coordinates of all maps around the locus would have to be changed.

3.1 Map Expression

First of all, objects in each fragmentary map should be addressed in a local coordinate system within the map, so that the specification of coordinates of objects does not need to be modified in the event that their island floats around. Namely, a relative addressing coordinate system is required. Next, for those fragmentary maps associated with some landmark, when the landmark moves around, they should follow without modification in their coordinate system.

In Lucy, a map representation language called an *map expression* has been introduced, which allows a map to be represented in a local coordinate system and in a relative addressing manner, and to be linked to another with an anchoring mechanism. The syntax of a map expression is defined as follows:

```
<MapExp> ::=   <Obj>
             | ':=' <MapExp>          | <MapExp> '=:'
             | <MapExp> ':<' <MapExp> | <MapExp> '<:' <MapExp>
             | '[' <MapExp> , ... , <MapExp> ']'
```

1. **Relative Addressing:** Two notions are associated with a map expression: one is the *current position* and the other is the *current direction*.

   (a) **Linear-Ordering**
   Expressions `A :< B` and `A <: B` mean, in common, that `A` is left of `B`; additionaly, the former means that `B` is evaluated after `A` is done, while the latter does that `A` is evaluated after `B` is done.

   (b) **Changing the Current Evaluation Direction**
   Expression `:= A` means to put the left bound of `A` at the current position and proceed the evaluation rightward, while expressin `A =:` means to put the right bound of `A` at the current position and proceed the evaluation leftward.

   (c) **Multi-Pinning**
   Expression `[A, B] <: C` means that `A` is left of `C` as well as `B` is left of `C`.

2. **Anchoring:** Objects constituting a map expression include positions and anchors (positions associated with labels). A label is globally accessible beyond a map expression so that it connects one map expression with another.

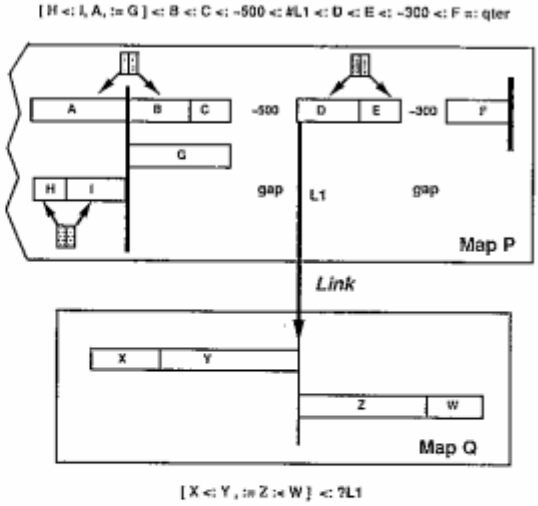

Figure 4: Map expressions

   (a) **Memorizing an Anchor**
   Expression `#L :< B` means to memorize the left bound of `B` under the label `L`.

   (b) **Referring to an Anchor**
   Expression `A <: ?L` means to refer to an anchor labelled with `L` to take it as the right bound of `A`.

Figure 4 illustrates an example of expressing two fragmentary maps, `P` and `Q`, which are linked up at the middle. Map `P` starts with the q-telomere which is followed by fragment `F`, a 300Kb gap, fragment `E`, fragment `D`, a 500Kb gap, fragment `C` and fragment `B`. At the left bound of fragment `B`, three other fragmentary maps start: one map proceeds pinning leftward on fragment `I` and then on fragment `H`, one map goes leftward from fragment `A`, and the other map goes rightward from fragment `G`. The position of the left bound of fragment `D` is labelled `L1` to be an anchor for map `Q`. Map `Q` contains two fragmentary maps starting with the anchor labelled `L1`. One map proceeds pinning with `Y` and then `X` leftward from the anchor, and the other does with `Z` and then `W` rightward from the anchor.

Figures 5, 6 and 7 show those maps represented in map expressions, using GenoGraphics.

Figure 5: this is an *Not*I restriction map around the q-telomere region of chromosome 21, some of whose data have been introduced in Table 1. *Not*I fragments and sites are shown in light green; gray lines denote hybridization signals between fragments and probes. Thus an interpretation of biological data is visualized to help understanding and verifying the mapping process.

Figure 6: in [34], regions are defined based on breakpoints (bounds) of various cell lines. The map of regions

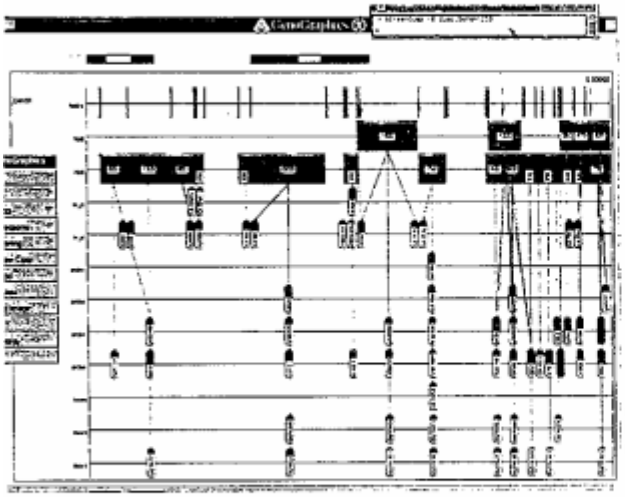

Figure 5: Visualizing a restriction map with hybridization signals

Figure 7: Three maps around locus D21S13

(labelled `Gar90`) is expressed with the breakpoints of the cell line panel (labelled `bkptst`) as anchors. For example, region `A7` is formed with the left bound of cell line `1;21` and the left bound of cell line `ACEM-2`.

Figure 7: three restriction maps, labelled `IchS13`, `CoxS13` and `Raf91`, are those around the `D21S13` locus whose position is given in the chromosome 21 physical anchor map (labelled `21phy`), scaled in percentage.

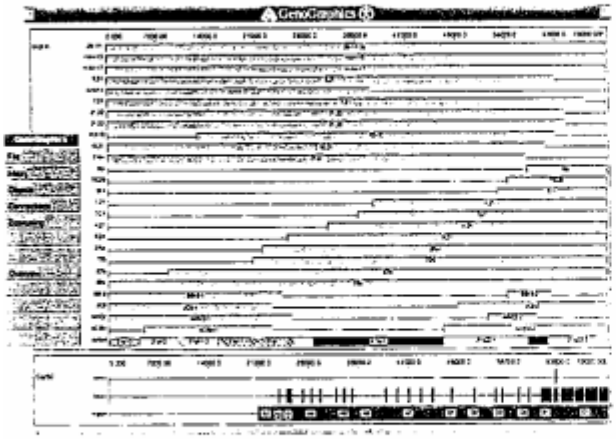

Figure 6: Gardiner's region map generated from a cell line panel

4 Inquiry to Lucy

This section presents the current status of queries Lucy can currently handle.

Concering the map visualized in Figure 5, the mapping effort started with the q-telomere and reached around locus D21S17 where a 1300Kb *Not*I fragment was pinned down. The following queries are issued to retrieve information related with this region so that the mapping effort can be advanced toward the centromere.

1. Regarding locus D21S17, its regional information is retrieved.

```
| ?- get_attributes(locus('D21S17'), [self(S)), within(Rs)]),
     print_object(Rs-S), fail.
------------------------------------------------------------------------
Object Id:
    -([region(21q22.1-q22.2)], GDB:locus(D21S17))
------------------------------------------------------------------------
Object Id:
    -([region(21q21.2-qter)], HGM10.5:locus(D21S17))
------------------------------------------------------------------------
Object Id:
    -([gardiner_region(B1), region(21q22.3)], LUCY:locus(D21S17))
------------------------------------------------------------------------
```

The region recorded in GDB is narrower than the one in HGM10.5 which is the predecessor of GDB. Also the D21S17 locus is assigned to region B1 in Gardiner's map shown in Figure 6.

2. Then, objects which occur left of D21S17 in all maps on which D21S17 occurs are retrieved.

```
| ?- setof(Obj-MID, Os^( occurs_on(map(MID), locus('D21S17')),
                         ordered_objects_on_map(map(MID), Os),
                         left_to(Obj, locus('D21S17'), Os) ),
          OMs),
     keymerge(OMs, KOMs), !,
     member(OM, KOMs), print_object(OM), fail.
-------------------------------------------------------------
Object Id:
    -(clone(pGSH8), [chr21_Denam1991_physical_around_21q22.3])
-------------------------------------------------------------
Object Id:
    -(gardiner_region(B1), [chr21_Gardiner1990])
-------------------------------------------------------------
Object Id:
    -(locus(D21S58), [chr21_Burmeister1991_RH, chr21_Petersen1991
    _female_meiosis, chr21_Petersen1991_male_meiosis, chr21_Tanzi
    1988_female, chr21_Tanzi1988_male, chr21_Tanzi1988_sex_averag
    ed, chr21_physical_anchors])
-------------------------------------------------------------
Object Id:
    -(locus(D21S82), [chr21_Warren1989_female_meiosis, chr21_Warr
    en1989_male_meiosis])
-------------------------------------------------------------
```

Beside clone pGSH8 and region B1, loci D21S58 and D21S82 are reported.

3. For the D21S58 locus, its regional information is retrieved.

```
| ?- get_attributes(locus('D21S58'), [self(S), within(Rs)]),
     print_object(Rs-S), fail.
-------------------------------------------------------------
Object Id:
    -([region(21q22.1-q22.2)], GDB:locus(D21S58))
-------------------------------------------------------------
Object Id:
    -([region(21q21)], HGM10.5:locus(D21S58))
-------------------------------------------------------------
Object Id:
    -([gardiner_region(D4)], LUCY:locus(D21S58))
-------------------------------------------------------------
```

Although the answers from GDB and HGM10.5 conflict, the locus is assigned to region D4 in Gardiner's map, which is to the left of region B1.

4. In order to grasp what more loci reside further left, all loci not only in region D4 but also in every D region are retrieved.

```
| ?- setof(R-Id, Rs^( get_attribute(locus(Id), within(Rs)),
                      member(gardiner_region(R), Rs),
                      substring(R, "D") ),
          RIs),
     keymerge(RIs, KRIs), !,
     member(KRI, KRIs), print_object(KRI), fail.
-------------------------------------------------------------
Object Id:
    -(D1, [D21S54])
-------------------------------------------------------------
Object Id:
    -(D2, [D21S93])
-------------------------------------------------------------
Object Id:
    -(D3, [D21S63, SOD1])
-------------------------------------------------------------
Object Id:
    -(D4, [D21S58, D21S65])
-------------------------------------------------------------
```

5. Finally, detailed information about locus D21S58 is retrived.

```
— ?- print'object(locus('D21S58')).
-------------------------------------------------------------
Categories:
   [1] locus

Input Date:
   1991/8/5

Investigators:
   [1] contact(P. C. Watkins)

Object Id:
   locus(D21S58)

Probes:
   [1] clone(524-SP)

References:
   [1] Kathleen Gardiner, Michel Horisberger, Jan Kraus, Umadev
   i Tantravahi, Julie Korenberg, Veena Rao, Shyam Reddy, David
   Patterson, "Analysis of human chromosome 21: correlation of p
   hysical and cytogenetic maps; gene and CpG island distributio
   ns", The EMBO Jounal, 9, 25-34, 1990

   [2] Michael B. Petersen, Susan A. Slaugenhaupt, John G. Lewis
   , Andrew C. Warren, Aravinda Chakravarti, Stylianos Antonarak
   is, "A Genetic Linkage Map of 27 Markers on Human Chromosom 2
   1", Genomics, 9, 407-419, 1991

Self:
   LUCY:locus(D21S58)

Within:
   [1] gardiner'region(D4)

[GDB/ver.1.0] Approved date:
   Sep  8 1990 10:57:11:140PM

[GDB/ver.1.0] Assignment modes:
   [1] somatic cell hybrids

[GDB/ver.1.0] Certainty:
   confirmed

[GDB/ver.1.0] Create date:
   Jun 18 1989  9:42:06:000AM

[GDB/ver.1.0] Full name:
   DNA Segment, single copy probe pPW524-SP

[GDB/ver.1.0] GDB:idx:
   GDB:locus(8242)

[GDB/ver.1.0] If cloned:
   yes

[GDB/ver.1.0] Information source:
   db(GDB/ver.1.0)

[GDB/ver.1.0] Input Date:
   1991/7/5

[GDB/ver.1.0] Locus type:
   DNA

[GDB/ver.1.0] Modify date:
   Nov 25 1990  2:01:47:640PM

[GDB/ver.1.0] Polymorphism type:
   polymorphic

[GDB/ver.1.0] Probes:
   [1] GDB:probe(pPW524-SP)

[GDB/ver.1.0] Report:
   include

[GDB/ver.1.0] Self:
   GDB:locus(D21S58)

[GDB/ver.1.0] Sources:
   [1] P. C. Watkins, R. E. Tanzi, J. Roy, N. Stuart, P. Stanisl
   ovitis, J. F. Gusella, "A cosmid clone genetic linkage map of
    chromosome 21 and localization of the breast cancer estrogen
   -inducible (BCEI) gene.", Cytogenet Cell Genet, 46, 712, 1987

   [2] M. Van Keuren, H. Drabkin, P. Watkins, J. Gusella, D. Pat
   terson, "Regional mapping of DNA sequences to chromosome 21."
   , Cytogenet Cell Genet, 40, 768-769, 1985

   [3] P. C. Watkins, P. A. Watkins, N. Hoffman, P. Stanisloviti
   s, "Isolation of single-copy probes detecting DNA polymorphis
   ms from a cosmid library of chromosome 21.", Cytogenet Cell G
   enet, 40, 775-774, 1985

   [4] M. L. Van Keuren, P. C. Watkins, H. A. Drabkin, E. W. Jab
   s, J. P. Gusella, D. Patterson, "Regional localization of DNA
    sequences on chromosome 21 using somatic cell hybrids.", Am
   J Hum Genet, 38, 793-804, Jun 1986

   [5] M. Burmeister, S. Kim, R. Price, T. de Lange, U. Tantrava
   hi, R. M. Myers, D. R. Cox, "A map of the long arm of human c
   hromosome 21 constructed by radiation hybrid mapping and puls
   ed-field gel electrophoresis", Genomics, In Press, ??, 1990

[GDB/ver.1.0] Symbol:
   D21S58

[GDB/ver.1.0] Within:
   [1] region(21q22.1-q22.2)
```

```
[HGM10.5] # of copies:
   single

[HGM10.5] Assignment modes:
   [1] somatic cell hybrids

[HGM10.5] Categories:
   [1] locus

[HGM10.5] Certainty:
   provisional

[HGM10.5] Information source:
   db(HGM10.5)

[HGM10.5] Input Date:
   1991/3/27

[HGM10.5] Probes:
   [1] clone(pPW524-SP)

[HGM10.5] References:
   [1] ref(Watkins et al (HGM8))

   [2] P. C. Watkins, R. E. Tanzi, K. T. Gibbons, J. V. Tricoli,
   G. Landes, R. Eddy, T. B. Shows, J. F. Gusella, "Isolation o
   f polymorphic DNA segments from human chromosome 21.", Nuclei
   c Acids Res, 13, 6075-88, Sep 1985

   [3] M. L. Van Keuren, P. C. Watkins, H. A. Drabkin, E. W. Jab
   s, J. F. Gusella, D. Patterson, "Regional localization of DNA
   sequences on chromosome 21 using somatic cell hybrids.", Am
   J Hum Genet, 38, 793-804, Jun 1986

   [4] ref(Nakai et al (HGM9))

[HGM10.5] Self:
   HGM10.5:locus(D21S58)

[HGM10.5] Within:
   [1] region(21q21)
```

Information are reported from publications, GDB and HGM10.5 in that order.

5 Concluding Remarks

Promoted by requirements in various application areas as well as in biology, steady progress in database technology has been made in the last few years [82].

Since the normal-form (1NF or *flat*) relational model [24] was proposed, practice over the years has pointed out its inefficiency in data access and its verbosity in inquiry [25, 28]. The source of both problems is the primitive data structure, the flat relation. In genome databases implemented upon normal form relational database systems, these problems are cast in relief, since the volume and variety of involved data are large and growing. In fact, the number of tables constituting a genome mapping database is apt to be quite large (e.g., 68 tables in LLNL Genome Database [4] and over 100 tables in GDB).

The present work could be regarded as one of the first to have successfully integrated public and local genome databases. The success greatly reflects the application of an object-oriented data representation and logic programming features, which should be the preliminary steps toward object-oriented databases [5, 36, 3, 7, 33, 85, 19] and deductive databases respectively. Through an experience with Lucy, it should be reasonable to conclude that these database technologies will contribute to the development and practice of genome databases.

Object-Oriented Database Technology. Since the notion of object-orientation [41] was invented in the field of programming languages, it has been widely disseminated over the past ten years [78, 91]. The heart of object-orientation, that is encapsulating the internal details of an object, is important for the implementation and retrieval of various kinds of data involved in genome databases. Lucy has only adopted an object-oriented data representation. Other ramparts have not been constructed yet: neither object-specific methods nor class inheritance. They will be future work.

Since the object-orientation was introduced to Lucy, some cases have been found where the framework does not fit naturally but where a nested (NF^2: non-first normal form) relational model [38, 1, 76] would. Here is an example. Given a table of linking clones, an entry for LA171 has been represented as follows:

name	region	# of occurrences MluI	BssHII	SacII	cloned fragments large	small
LA171	21q22.3	1	2	3	3.0	2.1
LA179	21cen	0	3	2	1.1	0.96
:	:	:	:	:	:	:

```
object('LA:clone'('LA171'),
      [input_date(1991/2/11),
       categories([linking_clone]),
       within([region('21q22.3')]),
       cloning_vector(lambda),
       linking_site(restriction_enzyme('NotI')),
       digested_from(genomic_DNA(human)),
       digested_by([restiction_enzyme('EcoRI')]),
       contains([times(restriction_enzyme('MluI'), 1),
                 times(restriction_enzyme('BssHII'), 2),
                 times(restriction_enzyme('SacII'), 3) ]),
       parts(['LA:clone'('LA171l'), 'LA:clone'('LA171s')]),
       references([ref('Saito et al (1991)')])
      ]).

object('LA:clone'('LA171l'),
      [input_date(1991/3/30),
       categories([half_linking_clone]),
       linking_site(restriction_enzyme('NotI')),
       size('Kb'(3.0))
      ]).

object('LA:clone'('LA171s'),
      [input_date(1991/3/30),
       categories([half_linking_clone]),
       size('Kb'(2.1))
      ]).
```

As shown in Table 1, `LA171l` and `LA171s` are those half linking clones which hybridized to fragments, 1800Kb and 750Kb, respectively. When these half linking clones were identified as objects, their sizes, 3.0Kb and 2.1Kb, were encapsulated in these objects. In contrast, consider the number of occurrences of restriction sites. It is questionable that an object should be created for the number of occurrences, such as once, twice or three times. Being part of an attribute, `contains`, the occurrences are stored as a nested relation of the form, `times/2`. For the third and forth columns in the example above, their relational structures are similar, but the meanings of their data imply different implementations. Further studies will be necessary to clarify this problem.

Deductive Database Technology. The necessity of loading an inference mechanism into a database system has been claimed in knowledge-intensive applications [16, 56, 63, 84].

Because most biological knowledge is symbolic rules on the four characters of DNA, there is a potential requirement for rule processing capability. A couple of genome database systems are being developed abreast of Lucy, exploiting logic programming facilities [42, 73, 6, 45]. In Lucy, the inference capability is being used mainly for query management. Few pieces of biological rules have been implemented.

References

[1] S. Abiteboul and N. Bidoit. Non first normal form relations to represent hierarchically organized data. In *Proceedings of the Third ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pp.191-200, Waterloo, April 1984.

[2] Serge Abiteboul and Paris C. Kanellakis. Object Identity as a Query Language Primitive. In *Proceedings of the 1989 ACM-SIGMOD Conference on Management of Data*, Portland, OR, May 1989.

[3] R. Agrawal and N. H. Gehani. ODE (Object Database and Environment): The Language and the Data Model. In *Proceedings of the 1989 ACM-SIGMOD Conference on Management of Data*, Portland, OR, May 1989.

[4] L. K. Ashworth, T. Slezak, M. Yeh, E. Branscomb and A. V. Carrano. Making the LLNL Genome Database 'Biologist Friendly'. *DOE Human Genome Program Contractor-Grantee Workshop*, Santa Fe, NM, February 17-20 1991.

[5] M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, S. Zdonik. The Object-Oriented Database System Manifesto. In *Proceedings of the First International Conference on Deductive and Object-Oriented Databases*, Kyoto, 1989.

[6] A. Baehr, G. Dunham, A. Ginsburg, R. Hagstrom, D. Joerg, T. Kazic, H. Matsuda, G. Michaels, R. Overbeek, K. Rudd, C. Smith, R. Taylor, K. Yoshida and D. Zawada. *An Integrated Database to Support Research on Escherichia coli.* Technical Report, Argonne National Laboratory, October 1991.

[7] J. Banerjee, H.-T. Chou, J. Garza, W. Kim, D. Woelk, N. Ballou and H. J. Kim. Data model issues for object-oriented applications. *ACM TOIS*, Jan 1987.

[8] F. Bancilhon and S. N. Khoshafian. A calculus of complex objects. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp.53-59. March 1986.

[9] Bart Barrell. DNA sequencing: present limitations and prospects for the future. *FASEB Journal*, 5:40-45, 1991.

[10] B. J. Billings, C. L. Smith and C. R. Cantor. New techniques for physical mapping of the human genome. *FASEB Journal*, 5:28-34 (1991).

[11] D. G. Bobrow and T. Winograd. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1:3-46. 1977.

[12] R. J. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171-216, April-June 1985.

[13] R. J. Brachman, A. Borgida, D. L. McGuinness, and L. A. Resnick. The CLASSIC knowledge representation system, or, KL-ONE: The next generation. *Workshop on Formal Aspects of Semantic Networks*, Santa Catalina Island, CA, February 1989.

[14] E. Branscomb, T. Slezak, R. Pae, D. Galas, A. V. Carrano and M. Waterman. Optimizing restriction fragment fingerprinting methods for ordering large genomic libraries. *Genomics*, 8: 351-66 (1990).

[15] Ivan Bratko. *Prolog: programming for artificial interlligence.* Second Edition, Addison Wesley, 1990.

[16] Bruce G. Buchanan and Edward H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, Addison-Wesley, 1984.

[17] Margit Burmeister, Suwon Kim, E. Roydon Price, Titia de Lange, Umadevi Tantravahi, Richard M. Myers, and David R. Cox. A Map of the Distal Regin of the Long Arm of Human Chromosome 21 Constructed by Radiation Hybrid Mapping and Pulsed-Field Gel Electophoresis. *Genomics*, 9:19-30, 1991.

[18] Paul Butterworth, Allen Otis and Jacob Stein. The GemStone object database management system. *Communication of the ACM*, Vol.34, No.10, pp.64-77, October 1991.

[19] Michael J. Carey, David J. DeWitt and Scott L. Vandenberg. A Data Model and Query Language for EXODUS. In *Proceedings of the 1988 ACM-SIGMOD Conference on Management of Data*, Chicago, IL, June 1988.

[20] A. V. Carrano. Estalishing the order of human chromosome-specific DNA fragments. *Basic Life Science*, 46: 37-49 (1988).

[21] A. V. Carrano, J. Lamerdin, L. K. Ashworth, B. Watkins, E. Branscomb, T. Slezak, M. Raff, P. J. de Jong, D. Keith, L. McBride. A high-resolution, fluorescence-based semiautomated method for DNA fingerprinting. *Genomics*, 4: 129-36 (1989).

[22] A. V. Carrano, P. J. de Jong, E. Branscomb, T. Slezak and B. W. Watkins. Constructing chromosome- and region-specific cosmid maps of the human genome. *Genome*, 31: 1059-65, (1989).

[23] M. J. Cinkosky and J. W. Fickett. SIGMA: Software for Integrated Genome Map Assembly. *Proc. of the 11th International Human Gene Mapping Workshop (HGM11)*, London, August 18-22, 1991.

[24] E. F. Codd, A Relational Model of Data for Large Shared Data Banks. *Communication of ACM*, 13:6, 1970.

[25] E. F. Codd. *The Relational Model for Database Management, Version 2.* Addison Wesley, 1990.

[26] A. Coulson, J. Sulston, S. Brenner and J. Karn. Toward a physical Map of the genome of the nematode, Caenorhabditis elegans. *Proc. Natl. Acad. ci. USA*, 83: 7821-7825, 1986.

[27] David R. Cox, Margit Burmeister, E. Roydon Price, Suwan Kim, Richard M. Myers. Radiation Hybrid Mapping: A Somatic Cell Genetic Method for Constructing High-Resolution of Mammalian Chromosomes. *Science*, 250: 245–250, 1990.

[28] C. J. Date. *An Introduction to Database Systems, Volume I*, Fifth Edition, Reading, Addison Wesley, 1990.

[29] Kay E. Davies and Shirley M. Tilghman, editors. *Genome Analysis Volume 1: Genetic and Physical Mapping.* Cold Spring Harbor Laboratory Press, 1990.

[30] *Human Genome 1989-90 Program Report.* DOE/ER-0446P, U.S. Department of Energy, March 1990.

[31] *Understanding Our Genetic Inheritance, The U.S. Human Genome Project: The First Five Years FY 1991-1995.* DOE/ER-0452P, U.S. Department of Energy.

[32] R. Durbin, S. Dear, T. Gleeson, P. Green, L. Hillier, C. Lee, R. Staden and J. Thierry-Mieg. Software for the C. Elegans Genome Project. *Genome Mapping and Sequencing Meeting*, Cold Spring Harbor Laboratory, NY, May 8-12 1991.

[33] D. Fishman et al. Iris: An object-oriented database management system. *ACM TOIS*, Vol.5, No.1, pp.48-69, January 1986.

[34] Katheleen Gardiner, Michel Hoisberger, Jan Kraus, Umadevi Tantravahi, Julie Korenberg, Venna Rao, Shyam Reddy and David Patterson. Analysis of human chromosome 21: coorrelation of physical and cytogenetic maps; gene and CpG island distributions. *The EMBO Journal*, vol.9, no.1, pp.25-34, 1990.

[35] D. Garza, J. W. Ajioka, D. T. Burke and D. L. Hartl. Mapping the Drosophila genome with yeast artificial chromosomes. *Science*, 246: 641-6 (1989).

[36] O. Deux et al. The O_2 system. *Communication of the ACM*, Vol.34, No.10, pp.34-48, October 1991.

[37] J. W. Fickett, M. J. Cinkosky, D. Sorensen and C. Burks. Integrated Maps: A Model and Supporting Tools. *Proc. of the 11th International Human Gene Mapping Workshop (HGM11)*, London, August 18-22, 1991.

[38] P. Fisher and S. Thomas. Operators for non-first-normal-form relations. In *Proceedings of the 7th International Computer Software Applications Conference*, Chicago, November 1983.

[39] Karen A. Frenkel. The Human Genome Project and Informatics. *Communication of ACM*, Vol.34, No.11, 40-51, 1991.

[40] *GDB and OMIM Quick Guide (Version 4.0).* GDB/OMIM User Support, The William H. Welch Medical Library, Baltimore, July 1991.

[41] Adele Goldberg and David Robson. *Smalltalk-80: The Language and Its Implementation.* Addison-Wesley, 1983.

[42] P. M. D. Gray and R. J. Lucas, editors. *Prolog and Databases: implementations and new directions.* Ellis Horwood, series in Artificial Intelligence, 1988.

[43] Ray Hagstrom. GenoGraphics. *International Chromosome 21 Workshop*, Denver, CO, April 10-11, 1991.

[44] L. Hood, R. Kaiser, B. Koop and T. Hunkapiller. Large-Scale DNA Sequencing. *DOE Human Genome Program Contractor-Grantee Workshop*, Santa Fe, NM, February 17-20 1991.

[45] Toni Kazic and Shalom Tsur. Building a Metabolic Function Knowledgebase System. *Workshop on Biological Applications in Logic Programming, ILPS'92*, San Diego, CA, November 1991.

[46] Y. Kohara, K. Akiyama and K. Isono. The Physical Map of the Whole E.coli Chromosome: Application of a New Strategy for Rapid Analysis and Sorting of a Lrge Genomic Library. *Cell*, 50: 495-508, 1987.

[47] M. Kosowsky, C. Blake, D. Bradt, J. Eppig, P. Grant, L. Mobraaten, J. Nadeau, J. Ormsby, A. Reiner, S. Rockwood, J. Saffer, T. Snell and M. Vollmer. Integration and Graphical Display of Genomic Data: *The Encycolopedia of the Mouse Genome. Genome Mapping and Sequencing Meeting*, Cold Spring Harbor Laboratory, NY, May 8-12 1991.

[48] Donald Johanson and Maitland Edey. *Lucy: The Beginnings of Human Kind.* Simon & Schuster, 1981.

[49] Peter Karp. *A Knowledge Base of the Chemical Compounds of Intermediary Metabolism.* unpublished, SRI International, September 1991.

[50] G. Kiernan, C. de Maindreville and E. Simon. Making Deductive Database a Practical Technology: a step forward. In *Proceedings of the 1990 ACM-SIGMOD Conference on Management of Data*, Atlantic City, NJ, May 1990.

[51] Charles Lamb, Gordon Landis, Jack Orenstein, Dan Weinreb. The ObjectStore database system. *Communication of the ACM*, Vol.34, No.10, pp.50-63, October 1991.

[52] Eric S. Lander and Michael S. Waterman. Genomic Mapping by Fingerprinting Random Clones: A Methematical Analysis. *Genomics*, 2:231-239, 1988.

[53] Eric S. Lander, Robert Langridge and Damian M. Saccocio. Mapping and Interpreting Biological Information. *Communication of ACM*, Vol.34, No.11, 32-39, 1991.

[54] Hans Lehrach, Radoje Drmanac, Jorg Hoheisel, Zoia Larin, Greg Lennon, Anthony P. Monaco, Dean Nizetic, Gunther Zehetner and Annemarie Poustka. Hybridization Fingerprinting in Genome Mapping and Sequencing. *Genome Analysis Volume 1: Genetic and Physical Mapping*, pp.39-81, Cold Spring Harbor Laboratory Press, 1990.

[55] A. J. Link and M. V. Olson. Physical map of the Saccaromyces cerevisia genome at 110-kilobase resolution. *Genetics*, 127: 681-98 (1991).

[56] Douglas B. Lenat and R. V. Guha. The Evolution of CycL, The Cyc Representation Language. *SIGART Bulletin*, Vol. 2, No. 3, ACM Press, June 1991.

[57] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley, 1990.

[58] S. Lewis, W. Johnston, V. Markowitz, J. McCarthy, F. Olken and M. Zorn. The Chromosome Inforamtion System. *DOE Human Genome Program Contractor-Grantee Workshop*, Santa Fe, NM, February 17-20 1991.

[59] Y. Lien. Hierarchical schemata for relational databases. *ACM Transactions on Database Systems*, Vol.6, No.1, pp.48-69, March 1981.

[60] David Lipman and James Ostel. Entrez: Sequences. In *Proceedings of the 11th International Human Gene Mapping Workshop (HGM11)*, London, August 18-22, 1991.

[61] Joseph Locker and Gregory Buzard. A dictionary of transcription control sequences. *DNA Sequence - Journal of DNA Sequencing and Mapping*, Vol.1, pp.3-11, 1990.

[62] Lohman et al. Extensions to Starburst: objects, types, functions and rules. *Communication of the ACM*, Vol.34, No.10, pp.94-109, October 1991.

[63] D. R. McCarthy and U. Dayal. The architecture of an active database management system. In *Proceedings of ACM SIGMOD 89*, pp.215-224, Portland OR, May 1989.

[64] A. Makinouchi. A consideration on normal form of not-neceissarily-normalized relation in the relational data model. In *Proceedings of the Third International Conference on Very Large Databases*, pp.447-453, Tokyo, October 1977.

[65] Victor A. Mckusick. Current trends in mapping human genes. *FASEB Journal*, 5: 12-20, 1991.

[66] H. W. Mohrenweiser, K. M. Tynan, E. W. Branscomb, P. J. de Jong, A. Olsen, B. Trask and A. V. Carrano. Development of an integrated genetic functional physical map of human chromosome 19. In *Proceedings of the 11th International Human Gene Mapping Workshop (HGM11)*, London, August 18-22, 1991.

[67] Fumio Mizoguchi. Knowledge Representation and Knowlegdge Programming. (in Japanese) *Computer Software*, Vol. 8, No. 4, JSSS, July 1991.

[68] D. Nelson and J. W. Fickett. An Electronic Laboratory Notebook for Data Management in Physical Mapping. *DOE Human Genome Program Contractor-Grantee Workshop*, Santa Fe, NM, February 17-20 1991.

[69] M. V. Olson, J. E. Dutchik, M. Y. Graham, G. M. Brodeur, C. Helms, M. Frank, M. MacCollin, R. Scheinman and T. Frand. Random-clone strategy for genomic restriction mapping in yeast. *Proc. Natl. Acad. Sci. USA*, 83: 7826-7830, 1986.

[70] Marnard Olson, Leroy Hood, Charles Cantor and David Botstein. A Common Language for Physical Mapping of the Human Genome. *Science*, 245: 28-29, September 1989.

[71] James Ostel. *GenInfo Backbone Database Overview (Version 1.0)*. Technical Report, NLM, NIH, Bethesda, MD, June 1990.

[72] *GenInfo ASN.1 Syntax: Sequences (Version 0.50)*. Technical Report, NCBI, NLM, NIH, Bethesda, MD, National Institutes of Health, MD, 1991.

[73] Norman W. Paton and Peter M. D. Gray. An object-oriented database for strage and analysis of protein structure data. In Reading [42].

[74] Mark L. Pearson and Dieter Söll, The Human Genome Project: a paradigm for information management in the life sciences. *FASEB Journal*, 5: 35-39, 1991.

[75] Peter Pearson. The Genome Data Base. *Proc. of the 11th International Human Gene Mapping Workshop (HGM11)*, London, August 18-22, 1991.

[76] Mark A. Roth and Henry F. Korth. The Design of ¬1NF Relational Databases into Nested Normal Forma. In *Proceedings of the 1987 ACM-SIGMOD Conference on Management of Data*, pp.143-159, San Francisco, May 1987.

[77] Akihiko Saito, Jose P. Abado, Denan Wang, Misao Ohki, Charles R. Cantor and Cassandra L. Smith. Construction and Characterization of a NotI Linking Library of Human Chromosome 21. *Genomics*, 10, 1991.

[78] John H. Saunders. A Survey of Object-Oriented Programming Languages. In *Journal of Object-Oriented Programming*, Vol.1, No.6, SIGS Publications, March/April 1989.

[79] H. Scheck and P. Postor. Data structures for an integrated data base management and information retrieval system. In *Proceedings of the Eighth International Conference on Very Large Databases*, pp.197-207, Mexico City, September 1982.

[80] "Genome Databases." *Science*, 254: 201-7 (1991).

[81] David B. Searls. *The Computational Linguistics of Biological Sequences*. Technical Report CAIT-KSA-9010, Center for Advanced Information Technology, UNISYS, PA, September 1990.

[82] Avi Silberschatz, Michael Stonebraker, Jeff Ullman (editors of the specicial issues on next-generation database systems). Database Systems: Achievements and Opportunities. *Communication of the ACM*, Vol.34, No.10, pp.110-120, October 1991.

[83] J. C. Stephens, M. L. Cavanaugh, M. I. Gradie, M. L. Mador and K. K. Kidd. Mapping the human genome: current status. *Science*, 250: 237-44 (1990).

[84] Michael Stonebraker et al. On rules, procedures, caching and views. In *Proceedings of the 1990 ACM-SIGMOD Conference on Management of Data*, Atlantic City, NJ, June 1990.

[85] Michael Stonebraker and Greg Kemnitz. The POSTGRES next-generation database management system. *Communication of the ACM*, Vol.34, No.10, pp.78-92, October 1991.

[86] J. Claiborne Stephens, Mark L. Cavanaugh, Margaret I. Gradie, Martin L. Mador and Kenneth K. Kidd. Mapping the Human Genome: Current Status. *Science*, 250: 237-250, 1991.

[87] Shunichi Uchida and Kaoru Yoshida. The Fifth Generation Computer Technology and Biological Sequencing. In *Proceedings of Workshop on Advanced Computer Technologies and Biological Sequencing*, Argonne National Laboratory, pp.28-36, November 1988.

[88] Denan Wang, Hong Fang, Charles R. Cantor and Cassandra L. Smith. A Contiguous NotI Restriction Map of Band q22.3 of Human Chromosome 21. to appear in *Proceedings of National Accademy of Science U.S.A.*, 1992.

[89] James Dewey Watson and Robert Mullan Cook-Deegan. Origins of the human genome project. *FASEB Journal*, 5: 8-11, 1991.

[90] James D. Watson, John Tooze, and David T. Kurtz. *Recombinant DNA – A Short Course –*. Scintific American Books, NY,1983.

[91] Kaoru Yoshida. *A'UM: A Stream-Based Concurrent Object-Oriented Programming Lanugage*. Ph.D thesis, Keio University, Japan, March 1990.

[92] Kaoru Yoshida, Cassandra L. Smith, Charles R. Cantor and Ross Overbeek, How will logic programming benefit genome analysis? *DOE Human Genome Program Contractor-Grantee Workshop*, Santa Fe, NM, February 17-20 1991.

[93] Kaoru Yoshida, Ross Overbeek, David Zawada, Charles R. Cantor and Cassandra L. Smith. Prototyping a Mapping Database of Chromosome 21. *Genome Mapping and Sequencing Meeting*, Cold Spring Harbor Laboratory, NY, May 8-12 1991.

[94] Kaoru Yoshida and Cassandra Smith. Key Features in Building a Physical Mapping Database System – Through an Experience of Developing a Human Chromosome 21 Mapping Knowledge Base System –. *The International Conference on the Human Genome (Human Genome III)*, San Diego, CA, October 21-23, 1991.

[95] David Zawada. *GenoGraphics for OpenWindows – v1.1 alpha*. Technical Report, Argonne National Laboratory, August 1991. (GenoGraphics is available via anonymous FTP from info.mcs.anl.gov (140.332.20.2),