Computer Note

A Prototype Object Database for Mitochondrial DNA Variation

J. E. NEIGEL AND P. LEBERG

From the Department of Biology, Box 42451, University of Louisiana, Lafayette, LA 70504. We thank Julie Waits for her assistance with literature searches. This research was funded by grant no. DBI9630311 from the National Science Foundation.

Address correspondence to Joseph E. Neigel at the address above, or e-mail: jneigel@louisiana.edu.

Surveys of biochemical and molecular genetic variation in natural populations have generated a wealth of data, but this valuable resource has not been adequately preserved. We hope to prevent further loss by establishing a community database for population genetic surveys. We explored the feasibility of a population genetics database by developing a prototype for animal mitochondrial DNA (mtDNA) surveys. This prototype includes the specification of a format for data files that are to be submitted to the database, an open-source object database that encapsulates data with methods to display and analyze data, and a website where data can be retrieved in either its original form or extensible markup language (XML). Data from more than 50 published surveys of mtDNA variation were retrieved from the literature and entered into the database. We hope that the population genetics community will support this project by contributing both data and expertise.

Since the introduction of allozyme methods in the mid-1960s there have been thousands of surveys of biochemical and molecular variation in natural populations (Leberg and Neigel 1999; Nevo 1988). These surveys have addressed problems in population genetics, systematics, ecology, and conservation biology (Avise 1994). Because of the breadth of these purposes, there has been little standardization in how genetic survey data are reported or archived. While flexibility in presentation is desirable for scientific publication, an unfortunate consequence is that much of the data are now lost or inaccessible. There is no central repository or database for population genetics data. Mining data from the literature is possible, but it is neither efficient nor reliable (Leberg and Neigel 1999). Without complete data, reanalysis or augmentation of population genetics surveys is unfeasible.

The scientific literature is the traditional forum for sharing scientific results (Johns 2002) and offers important benefits. However, anyone who has attempted to systematically retrieve data from the literature is likely to have encountered its limitations as a data archive. Datasets that require more than a few journal pages are seldom printed in full (Leberg and Neigel 1999) and new methods of genetic analysis will generate even larger quantities of data. We are not suggesting that scientific publication is obsolete, but only that it should not be considered an effective mechanism for archiving or disseminating data.

We explored the feasibility of a database for population genetic surveys by developing a prototype, the Population Genetics Database (PGDB). For practical reasons, we limited the prototype to surveys of animal mitochondrial DNA (mtDNA) variation. The physical structure of the mitochondrial genome and its transmission genetics are consistent across most animal taxa (Moritz et al. 1987) and can be represented objectively (Leberg and Neigel 1999). Although different types of data are used to represent mtDNA variation, they are comparable for estimates of genetic diversity or divergence. When we began this project in 1996, we determined there were about 600 published surveys of mtDNA (Leberg and Neigel 1999).

Flat File Database

Population genetics data come in many forms, and technological advances continue to introduce new types of data. It is therefore essential that a community population genetics database accommodate diverse forms of data and provide a simple mechanism for the addition of new types. Furthermore, a community database is unlikely to be successful if the process of data submission is too burdensome. A simple solution would be to submit data files that had already been prepared for population genetics analysis packages. However, none of the formats that are currently in use for these packages include the contextual information that is needed to evaluate and compare population genetics surveys. For example, these formats generally do not include the geographic locations of samples, the names of restriction endonucleases associated with particular sites or fragments, accession numbers for sequence databases, or citations of publications. Extensible markup language (XML) could be used to represent this information, but it is difficult to prepare XML files by hand. We therefore developed the PGDB flat file format to serve as a mechanism for data entry that is both flexible and ergonomic. The format specifies plain text files with minimal use of specialized punctuation. Keywords placed before individual values, rows, or tables identify the types of data that follow. Single and double line breaks are used to separate groups of data items, although very long items such as DNA sequences can be joined over multiple lines. This format should allow database entry files to be easily prepared by hand, often with only slight modification of data files that have been prepared for other purposes. A complete specification of the format can be found on the PGDB website (http://seahorse.louisiana.edu/PGDB/).

The PGDB flat file format can be used to archive data in a form that is both human and machine readable. The format is open; keywords can be added to specify new types of data and aggregate data types can be defined by combinations of keywords. The price of this flexibility is that computer programs written to parse these files will need to interpret many possible arrangements of data. In practice, however, it is possible to limit this format for particular purposes. We tested this approach by developing a format for the entry of data from mtDNA surveys into our prototype database; an example is shown in Table 1. A collection of these input files constitutes a flat file database that can be managed by any operating system that provides for the storage and retrieval of text files. The PGDB website can be used to search these files for arbitrary text; each file that has been found to contain the text is listed with a hyperlink. Although not a true database management system (DBMS), this simple collection of text files serves as a data archive, much like the original GenBank database. These text files can also serve as input for more sophisticated databases and do not limit the DBMS to a single choice.

Object Database

Although most commercial databases follow the relational model (Codd 1982), we decided to experiment with the newer object model (Loomis 1990) for our prototype. Object databases offer particular advantages in scientific applications and have been developed for molecular biological data (Gray et al. 1990; Kemp and Gray 1990; Kochut et al. 1993; Shin et al. 1992). Although most large commercial databases are relational, the world's largest database is an object database that collects massive amounts of data from high-energy physics experiments (Adesanya et al. 2003). Object databases are based on object-oriented programming languages such as C++ and Java (Jordan 1998). Objects represent "behaviors" of entities as well as their attributes, and these behaviors can include methods to transform, analyze, and display data. Both the types of data that represent the attributes of objects and the methods that perform behaviors are defined for classes of objects. Polymorphic classes overcome the problem of data heterogeneity by defining analogous methods for objects that are conceptually similar, but based on different types of data. For example, a method to estimate sequence divergence could be implemented both for objects that represent restriction site data and for those that represent sequence data. This would allow both types of objects to be members of a class of objects that represents genotypes. Methods supplied by classes also simplify the development of database applications. Simple applications can be based almost entirely on these methods, while the effort needed to develop more complex applications is greatly reduced.

Table 1. Simple example of data from a single mtDNA survey that has been formatted for entry into the PGDB database

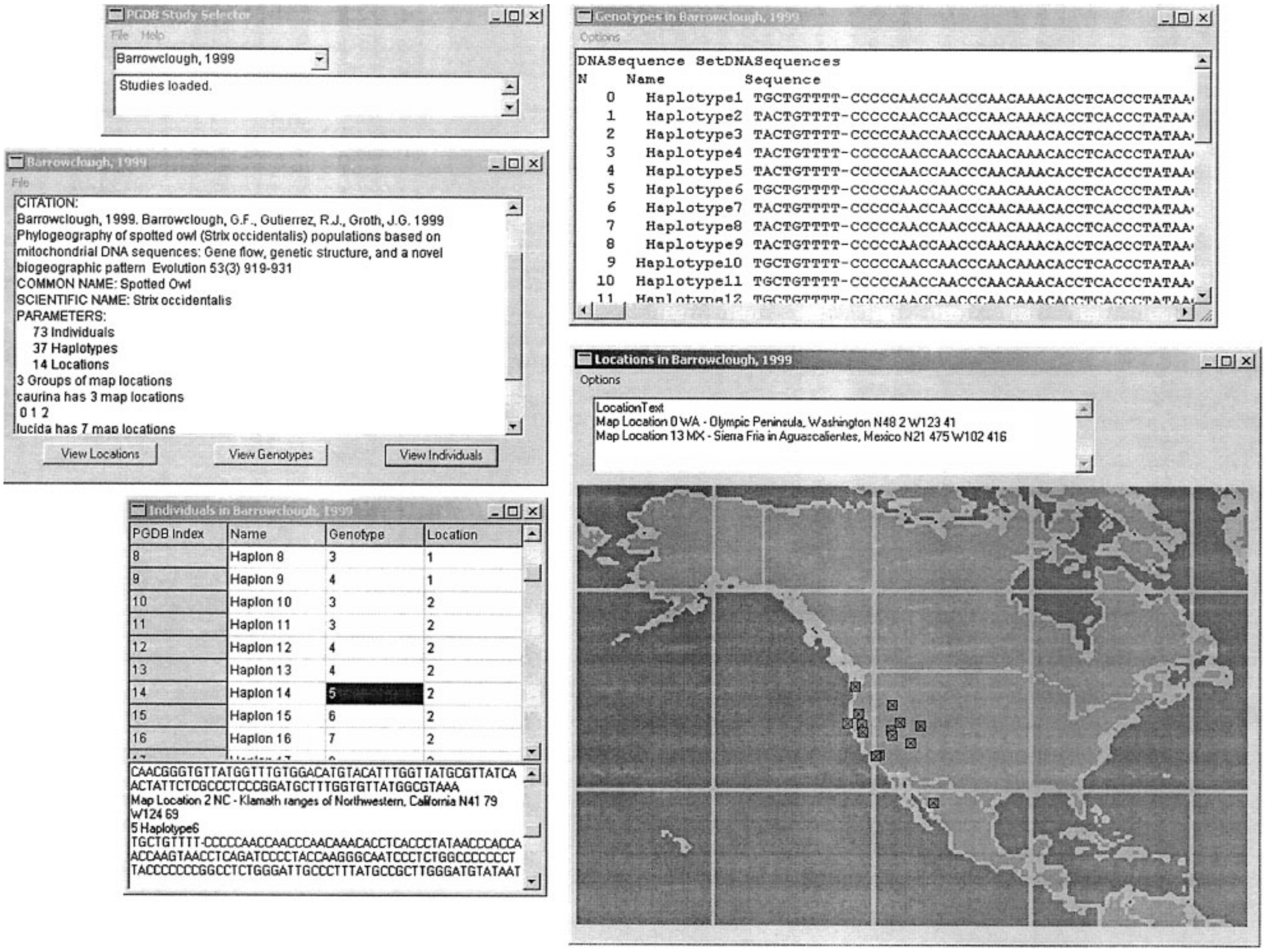
```
Name Smith, 2002
Comment: This is not a real data file, its only an example.
Columns GenotypeLabel DNASequence
“Northern Haplotype” GTCTATTTGAAGATATAAATAGTCT
“Southern Haplotype” GTCTGTTTAGGAGTATAAATAGTCT

Columns LocationLabel LatitudeDegrees LongitudeDegrees Repeat 2 GenotypeCounts
“North Pole” N90 E0 10 0
“South Pole” S90 E0 0 10
```

Fundamental Classes and Relationships

Most population genetics studies are alike in providing data on the characteristics of three basic entities: individuals, locations, and genotypes. In our object database, each of these is represented by a base class; classes derived from them represent specific types. For example, from the base class that represents genotypes, one class was derived for genotypes that are defined simply as numbered haplotypes, another class for genotypes that correspond to DNA sequences, and a third for those defined by restriction analysis. From the latter, two additional classes were derived: one for restriction fragment data and the other for restriction site data. All genotype classes have methods to estimate parameters of genetic diversity and divergence. Classes were derived from the base class for locations representing Cartesian (xy) coordinates and spherical (latitude and longitude) coordinates. These classes have methods for displaying locations on maps and calculating distances. Another class of objects represents individual organisms; each individual is linked to a genotype and location. An interrelated set of individual, location, and genotype objects is managed by an object that represents a population genetics study. The study object also manages data that define the context of the study, such as bibliographic citations and the common and scientific names of the organism. The most common types of data from mtDNA surveys are represented by classes in the PGDB object database, with more than 350 methods to manage, display, and analyze these data. For types of data or methods of analysis not represented by these classes, new classes and methods can be easily added.

Platform Independence and Database Management

We feel strongly that a community population genetics database should be built with open source software and not be tied to any specific DBMS, operating system, or computer platform. The core code of the PGDB object database is written in standard C++ and is distributed under an open source license agreement. We have also included an interface class to serve as a bridge between PGDB objects and the graphical user interface (GUI) methods of specific operating systems. For each GUI, these methods can be implemented without altering the core code. An interface class was also implemented for console methods from the standard C++ input/output (I/O) library. Applications that use interface classes should work with all existing PGDB classes, as well as any new classes that are derived from them.



Studies loaded.

CITATION:
Barrowclough, 1999. Barrowclough, G.F., Gutierrez, R.J., Groth, J.G. 1999
Phylogeography of spotted owl (Strix occidentalis) populations based on
mitochondrial DNA sequences: Gene flow, genetic structure, and a novel
biogeographic pattern Evolution 53(3) 919-931
COMMON NAME: Spotted Owl
SCIENTIFIC NAME: Strix occidentalis
PARAMETERS:
  73 Individuals
  37 Haplotypes
  14 Locations
3 Groups of map locations
caurina has 3 map locations
 0 1 2
lucida has 7 map locations

```
DNASequence SetDNASequences
N     Name          Sequence
   0   Haplotype1 TGCTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   1   Haplotype2 TACTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   2   Haplotype3 TACTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   3   Haplotype4 TACTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   4   Haplotype5 TACTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   5   Haplotype6 TGCTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   6   Haplotype7 TACTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   7   Haplotype8 TACTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   8   Haplotype9 TACTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
   9  Haplotype10 TGCTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
  10  Haplotype11 TGCTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
  11  Haplotype12 TGCTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAA
```

LocationText
Map Location 0 WA - Olympic Peninsula, Washington N48 2 W123 41
Map Location 13 MX - Sierra Fria in Aguascalientes, Mexico N21 475 W102 416

PGDB Index	Name	Genotype	Location
8	Haplon 8	3	1
9	Haplon 9	4	1
10	Haplon 10	3	2
11	Haplon 11	3	2
12	Haplon 12	4	2
13	Haplon 13	4	2
14	Haplon 14	5	2
15	Haplon 15	6	2
16	Haplon 16	7	2

CAACGGGTGTTATGGTTTGTGGACATGTACATTTGGTTATGCGTTATCA
ACTATTCTCGCCCTCCCGGATGCTTTGGTGTTATGGCGTAAA
Map Location 2 NC - Klamath ranges of Northwestern, California N41 79
W124 69
5 Haplotype6
TGCTGTTTT-CCCCCAACCAACCCAACAAACACCTCACCCTATAACCCACCA
ACCAAGTAACCTCAGATCCCCTACCAAGGGCAATCCCTCTGGCCCCCCCT
TACCCCCCCGGCCTCTGGGATTGCCCTTTATGCCGCTTGGGATGTATAAT

Figure 1. Screenshot of the PGDB browser application, which uses a Microsoft Windows implementation of a PGDB interface object. A database has been loaded and the data from a single mtDNA sequence survey are displayed. The window that displays the sequences can be used as an alignment editor. Clicking on a map location displays its name and coordinates in the text box above the map. The table in the lower left lists the genotype and location of each individual; clicking on the corresponding cell provides details.

The PGDB object database prototype includes a simple DBMS to create, store, and retrieve databases. We recognize that a more complete DBMS will be needed in the future, but we included this basic functionality to avoid using a proprietary DBMS in the prototype. For many purposes, a full-featured proprietary DBMS is unnecessary and would compromise our goal of building the prototype with open source software. For example, locking protocols would be needed to negotiate concurrent attempts by multiple users to modify the same data; but there is no such need if data are archived and not regularly altered. Reorganization of data into tables for a relational DBMS (object-relational mapping) would provide more efficient access to individual data items, but would offer little advantage if it is more useful to work with complete objects that represent individual surveys. We expect that the best choice for a DBMS will become more clear from experience with the prototype.

The recent emergence of XML as a standard for data on the Internet suggests a new possibility for the management and dissemination of population genetics data. XML figures prominently in plans for a "Semantic Web" that will facilitate the transfer of scientific data both within and between disciplines (Berners-Lee and Hendler 2001). XML tags specify the types of data found in a document rather than just how it should be formatted. XML documents can be displayed by compatible web browsers or stored and manipulated in databases (Bray et al. 2000). One advantage of XML is that it can be managed by relational databases, object databases, or native XML databases. We defined a set of XML tags for the data in our prototype and used methods in the object database to generate an XML document for each survey; these documents can be retrieved from the PGDB website.

Browser Application

Because object databases offer methods for data manipulation and display, it is easy to build applications with these methods. We developed a database browser that demonstrates how the PGDB object database can be used to develop an interactive, graphical application. The browser was developed specifically for 32-bit Microsoft Windows operating systems with the rapid application development tool Borland C++ Builder (version 5.0). It includes an interactive data parser in which one window acts as a text editor for the input file, while a second window displays how the parser has interpreted each line. This arrangement allows the input file to be quickly corrected and reparsed. The browser can also be used to enter, retrieve, and display data with dropdown menus, list boxes, bitmaps, and other graphical interface elements (Figure 1).

A Database of Methods

A long-term goal of the PGDB project is to facilitate the dissemination, testing, and application of data analysis methods. We intend the PGDB to serve as a repository for data analysis methods as well as provide a framework to support the development of new data analysis programs. Much of the code that is written for data analysis software is redundant; parsing data from input files, calculating basic statistics, and interacting with the user. Often the amount of this supporting code exceeds that needed to perform the actual analysis. The PGDB provides a core of common functions and thus facilitates the creation of easily used, cross-platform population genetics data analysis software. With this core functionality, it is possible to write useful applications with only a few lines of code. To encourage the development of PGDB extensions and applications, the complete source code for the PGDB has been made available on the project's website, along with extensive documentation on how to write PGDB programs. Under the terms of the open-source general public license that apply to the PGDB source code, software based on PGDB code is also open source.

Prospects

The PGDB prototype is intended to serve as a working model for a community population genetics database. Not everyone will agree with our design decisions, but we hope that most will recognize that it is important to capture the raw data of population genetics surveys that are now being lost. We are aware of some of the potential downsides to the establishment of a community database. Some investigators have expressed concerns about releasing their data before they have fully explored their implications or before they have completed long-term projects from which they expect multiple publications. We are sympathetic to these concerns, but we also believe that data used to support published scientific findings should be available for scrutiny. The establishment of a database does not by itself create an obligation to provide data; such requirements are established by individual journals. We therefore encourage investigators to submit data voluntarily, except in cases where privacy or proprietary claims would be threatened.

Both formal and informal polling have indicated that there is support within the population genetics community for a database. In 1993, J. E. Neigel sent a questionnaire to the editors of journals that frequently publish population genetics data. The majority of the respondents indicated that a database should be established, that the database should be supported by a federal agency, and that it would significantly reduce publication costs. We have also received overwhelmingly positive responses at national and international meetings where we have presented our proposal to establish a database. We now invite our colleagues to contribute, critique, and otherwise participate in the future development of a Population Genetics Database.

References

Adesanya A, Azemoon T, Becla J, Hanushevsky A, Hasan A, Kroeger W, Trunov A, Wang D, Gaponenko I, Patton S, and Quarrie Det al., 2003. On the verge of one petabyte: the story behind the BaBar database system. In: Computing in high energy and nuclear physics 2003. La Jolla, CA: University of California, San Diego; 1–6.

Avise JC, 1994. Molecular markers, natural history and evolution. New York: Chapman & Hall.

Berners-Lee T and Hendler J, 2001. Publishing on the semantic web: the coming Internet revolution will profoundly affect scientific information. Nature 410:1023–1024.

Bray T, Paoli J, Sperberg-McQueen CM, and Maler E, 2000. Extensible markup language (XML) 1.0, 2nd ed. World Wide Web Consortium.

Codd EF, 1982. Relational database: a practical foundation for productivity. Commun ACM 25:109–117.

Gray PM, Paton NW, Kemp GJ, and Fothergill JE, 1990. An object-oriented database for protein structure analysis. Protein Eng 3:235–243.

Johns A, 2002. The birth of scientific reading. Nature 409:287.

Jordan D, 1998. C++ object databases: programming with the ODMG standard. Reading, MA: Addison-Wesley Longman.

Kemp GJ and Gray PM, 1990. Finding hydrophobic microdomains using an object-oriented database. Comput Appl Biosci 6:357–363.

Kochut KJ, Arnold J, Miller JA, and Potter WD, 1993. Design of an object-oriented database for reverse genetics. In: Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, Bethesda, MD, USA, July 1993 (Hunter L, Searls DB, and Shavlik JW, eds.). AAAI/MIT Press; 234–342.

Leberg PL and Neigel JE, 1999. Enhancing the retrievability of population genetic survey data? An assessment of animal mitochondrial DNA studies. Evolution 53:1961–1965.

Loomis MES, 1990. ODBMS vs relational. J Object Orient Prog 3:79–82.

Moritz C, Dowling TE, and Brown WM, 1987. Evolution of animal mitochondrial DNA: relevance for population biology and systematics. Annu Rev Ecol Syst 18:269–292.

Nevo E, 1988. Genetic diversity in nature: patterns and theory. Evol Biol 23:217–246.

Shin DG, Lee CH, Zhang JH, Rudd KE, and Berg CM, 1992. Redesigning, implementing and integrating Escherichia *coli genome* software tools with an object-oriented database system. Comput Appl Biosci 8:227–238.

Received June 27, 2002
Accepted October 20, 2003

Corresponding Editor: Robert Fleischer
DOI: 10.1093/jhered/esh015