Data Note System for Capturing Laboratory Data

Mark Graves

Department of Cell Biology, Baylor College of Medicine
One Baylor Plaza, Houston, TX 70030

New Correspondence Address:

Mark Graves
Mercator Genetics
4040 Campbell Ave.
Menlo Park, CA 94025

mgraves@mercator.com (e-mail)
(415) 614-7010 (voice)
(415) 617-0883 (fax)

Subject Category: New technology and resources

Running Head: Data Note System

Abstract

The complexity of genome data limits the usefulness of traditional database management systems. The highly interconnected structure of genome data can be captured in a data representation language based on the mathematical formalism of graphs. We have tailored graphs for describing genome data, and have developed a database management system, called the Data Note System, for developing small databases to capture data from genome laboratories.

To simplify the use of the Data Note System, a series of tools with graphical user interfaces have been developed. The system is designed to be easy to install and use by novice database developers with a minimal amount of computer expertise. We describe the tools and present examples of their use. The system consists of a storage facility, a schema editing tool to simplify the design of small databases, and three tools for data entry and querying.

Advances in laboratory automation and experimental techniques are generating an increasing amount of laboratory data. Previously, this data would have been stored in laboratory notebooks, but the amount of data has grown to where laboratory notebooks are insufficient for data capture. Some researchers have taken the step to store the information electronically in spreadsheets. However, spreadsheets are limited by the amount of data they can efficiently contain and the lack of query capabilities. Although spreadsheets can be manually searched for values or combined with other spreadsheets, the user is soon left with a large number of spreadsheets containing the data in various formats which rapidly becomes out of date.

Another approach is to develop databases. Several small, file-based database management systems are available, such as 4D or FoxPro, which provide query capabilities and storage facilities that are not available in spreadsheets. In addition, many larger commercial database management systems are available to capture even larger quantities of data. However, the problem facing many would-be designers of biology databases is not capturing the quantity of data in a database, but capturing its complexity. The complexity of biology data is not easily captured in the fixed record format of traditional database systems. As more complex data is to be captured, the biologist developing the database is forced either to address the rising complexity by becoming immersed in database design techniques or to turn the database over to a database developer who frequently does not understand the biology. If a very large amount of data is to be captured, there is often no other choice but to rely on experienced database developers, and techniques have been developed to make this interdisciplinary design process work more smoothly [Graves *et al.* 1996]. However, for the amount of data typically generated by research laboratories, there is another solution.

The Data Note System is a system designed to capture data from biology research laboratories and represent it in a form which is easy to store and query. It grew out of a realization that there

was an increasing need for databases to capture biological information, and that bioinformatics software development efforts were typically heading in the direction of larger, more complicated systems which were beyond the capabilities of most small laboratories to install, use, and maintain. The system is designed for ease of installation and database development, allowing a novice database developer to implement a laboratory database with a minimal amount of database expertise. It also provides a fairly sophisticated query facility to encourage exploration of the captured data.

To develop a database using the Data Note System, a user:

1. sketches out (on paper) the concepts to be captured (as illustrated in Figure 1);
2. uses a schema editing tool to develop a template for each concept (as illustrated in Figure 2);
3. adds data to the database using a data entry tool created automatically from the template (as illustrated in Figure 3); and
4. retrieves data by asking questions using either of a tool (as illustrated in Figures 4) or a data exploration tool (as illustrated in Figure 5 for a different laboratory example).

Because biological laboratory data changes rapidly, the system is designed to grow as the data changes. Automatic generation of data entry and query tools from the schema reduces the amount of development time necessary.

System Description. A database in the Data Note System consists of a text file that contains data stored as graphs. A labeled edge of a graph is used to describe one characteristic of an entity or relationship, such as the name of a person or the strength of a hybridization reaction. Each line

in the file contains one edge, which consists of a source vertex, an edge label, and a destination vertex. The use of graphs as a foundation for the data model of the Data Note System:

- allows new, complex biological concepts to be built up incrementally from existing ones by adding new relationships as needed,
- simplifies the introduction of many-to-many relationships into the database,
- limits the overhead required for developing the initial database and maintaining the database, and
- simplifies the representation of data to a form which is easier to manipulate computationally and manually.

Small laboratory databases can be developed to capture a variety of data types including information about reagents, such as clones. Information about clones used in a laboratory can be captured as a conceptual schema which are used to define what data “looks like” in a database. A *schema* describes the concepts and relationships which compose the data in a database. The simplest useful description of a clone might contain only the name and type of a clone (such as YAC, cosmid, or M13), though a more complete schema might contain information about a clone’s content, construction, and storage. A possible schema for plasmid laboratory data is given in Figure 1. The schema for plasmid consists of a central concept to which other concepts are related. The concepts are linked via labeled edges which describe a characteristic of the plasmid. For example, the name of a plasmid is captured in the schema by an edge labeled name which links the plasmid concept with the name concept. Additional information can also be added to schemas, such as the restriction that there are possibly many “cassettes” but only one date that the entry was added to the database.

The conceptual schema is broken up into subgraphs called graph templates. Graph templates describe the types of data that can be stored in a database. The intention behind the Data Note System is to allow the developer as much flexibility as possible at this stage in database design.

The database developer edits each template in the schema by using a template editing tool to draw a template. Each template defines the vertices and edges which are to be created for each type of data to be included in the database. Edges described by the graph are created when a data item of this type is being added to the database. Each graph template can be refined and extended in isolation keeping the other templates constant. For the plasmid example, the graph template consists of one graph which covers the entire conceptual schema. For example, a “plasmid” graph template would have edges for the name, cassette, backbone, promoter, description, lab locations, constructor, and date entered, as shown in Figure 2. Graph templates are created within the edit template tool which consists of (a) an area to create three kinds of vertices: parameters, hidden nodes, and constants, (b) an area to select or create edge labels, (c) a series of command buttons, and (d) an area in which to draw the template.

After the schema is created, the user can enter data in an automatically-generated data entry form, much as it could be entered into a spreadsheet. The data entry tool uses the graph templates to create a graphical user interface in which the user enters data. The data entry forms are created automatically from the templates in the schema, and if the schema changes, the data entry tools change, too. The automatic creation of data entry forms allows the database to be rapidly developed and to be modified when the laboratory process changes. Figure 3 shows the data entry tool for entering plasmid data. (An additional tool not described here, but available in the distribution of the Data Note System, transfers data from an existing spreadsheet into the Data Note System to prevent a user from having to manually re-enter data.)

Capturing laboratory data is the first step in many processes. The data can later be analyzed, manipulated, and explored. Two likely uses of laboratory data are to obtain experimental results which support or disprove the original hypothesis and to guide future experiments by making use of previously captured data. Two tools are provided in the Data Note System to support querying and data exploration: a query-by-example tool and an *ad hoc* query tool.

Query-by-example is a query paradigm where the user asks a question by specifying part of the data in a template, and the query tool fills in the rest of the template based on the existing data. The query-by-example tool creates a graphical user interface for each graph template, as is done in the data entry tool. The user enters data into part of the template, and the system generates a report of all the data which matches the partially-filled template.

The query-by-example tool is shown in Figure 4 for the plasmid laboratory data template. For example, to find information about a specific plasmid, the user selects the entry box labeled “name”, types the name, and clicks on the “Query” button. This brings up a separate window which contains the report which shows the information. Different reports can be created by specifying different fields in the query-by-example form, for example, showing all plasmids with a specified cassette. Reports are created as tab-delimited text files which can be displayed using the Data Note System, exported into a spreadsheet, included as a table in a document, or further manipulated using external tools.

Sometimes the user may want to see a report which does not fit an existing graph template. These reports are created using the *ad hoc* query tool. The *ad hoc* query tool provides the user a flexible graphical user interface:

- to create queries which were not considered when the database was originally defined,
- to create queries which combine characteristics of multiple concepts,

- to incrementally create complex queries for data retrieval and exploration, or
- to experiment with queries on the data to discover novel relationships.

The user creates a temporary graph template using the *ad hoc* query tool and receives a report of all matching data. The user creates an *ad hoc* query in a similar fashion to how a permanent graph template is created using the schema design tool. A laboratory example where *ad hoc* queries are useful is a genotyping laboratory, where data discovery queries of genotyping data can be used for genetic analysis. In genetic analysis, the programs usually require information such as, the alleles at each marker for which individuals were typed, the family to which each individual belongs, the individual's position in the family, and the affected status of each individual. This information can be used to define a region of the genome where a gene contributing to a specific disease may reside. Figure 5 shows a query where the typing, family, and diagnostic data are combined to retrieve information which would be useful for genetic analysis.

Discussion. It is difficult to design a database for genome and biological data because of the complexity of the data and related concepts. Many significant biological database projects to date have been implemented using relational database technology because the volume of data and the need to assure its security requires the use of mature, commercial database management systems. This places additional constraints on the design of a genome database because relational databases are optimized for transaction-intensive, record-oriented applications; not applications involving data that has a complex structure. We have developed a database system which is more natural for capturing genome data and its rapidly changing requirements.

Larger database systems have been developed to capture genome data. OP/M [Chen and Markowitz 1994] captures objects and protocols of biological experiments; it has been implemented using a relational database by translating each OP/M schema into a relational one. Lab-

Base [Goodman *et. al.* 1995] has been developed to model biological experiments, and Genome Topographer [Marr unpublished] is a system designed to facilitate data publication and release of genome maps and sequence. These are not designed or intended to be easily installed, used or maintained by novice database developers with limited resources. ACeDB [Durbin and Thierry-Meig 1991] is a stand-alone system developed to provide data persistence for genome activities which has been used with success. However, ACeDB does not currently provide an extensible framework which allows for extension by novice database developers nor is it clear that its linked-tree data structure is sufficient to capture the wide variety of biological data [Mirkin and Rodin 1984].

One small database system cannot capture all the types of data from a laboratory. The Data Note System is designed to capture symbolic and numeric data which contains primarily structural relationships, such as maps or pathways, and measurements, such as weights, sizes, and data from gels. It supports data that might be presented as a table, and adds flexibility to the representation of that data. It captures archival data well, such as the results of multiple experiments; this is particularly useful to avoid duplication of experiments when large numbers of results are generated in a semi-automated fashion. It also provides data entry tools which closely match the users description of the data. The system does not capture pictorial data, such as the images of gels, or have built-in arithmetic operations, such as those a spreadsheet or statistical database might have. It does not support modifying data fields, such as counters or status indications. The system also is not designed to capture large amounts of sequence data.

The Data Note System is currently designed to store graphs with approximately 10,000 to 30,000 edges with no noticeable slowdown in performance. Graphs one to two orders of magni-

tude larger have been used, but the queries can take several seconds to a few minutes on a Sun workstation.

Advantages of the Data Note System over large, commercially-available database systems, such as Oracle, Sybase, or Gemstone, are: ease of incrementally developing small databases, ease of exporting the data and accessing it from stand-alone applications, cost, simpler maintenance, and ease of use. The advantage of larger database systems is the speed of access for large databases. A disadvantage of larger database systems is the unnecessary overhead inherent in the larger systems. Large database systems are often designed for simultaneous access by thousands of users in multiple geographic locations. They must operate with no down time in an environment where hostile users may try to crash the system or gain unauthorized access. Multiple groups or departments in the organization may have to approve each schema change, and protecting the data from inappropriate use is more important than allowing the data to be easily accessed. Small research laboratories do not typically deal with these issues and do not need the concomitant overhead provided in the large systems.

Advantages of the Data Note System over small relational data systems, such as 4D, include the ease of database design and use. Although 4D also has a graphical user interface, the user must still learn relational database design techniques, and it is difficult to design databases with a complex schema. It is also difficult for end users to make *ad hoc* queries. An advantage of 4D over the Data Note System is the restricted data entry provided by 4D data entry tools which restrict the types of data which can be entered into a specific field.

Software installation. The Data Note System is available via anonymous ftp from the machine stork.bcm.tmc.edu in the directory /pub/dns1.3/ (Internet address 128.249.6.20). More information about the Data Note System is available at http://stork.bcm.tmc.edu/dns/ including an

overview of the system, a tutorial, an on-line user manual, example data files, and installation information. The on-line documentation provides information to allow a novice database developer to develop simple databases.

The Data Note System is implemented in Tcl/Tk (version 7.6/4.2) which is required for it to be installed and used. Tcl (tool command language) is a scripting language, and Tk is a graphical user interface tool kit based on Tcl. Both packages are freely available from Sun Microsystems. More information about Tcl/Tk, including information on obtaining it via anonymous ftp, is available at http://www.sunlabs.com/research/tcl/. The Data Note System requires 350K of disk space, and the Tcl/Tk executables require an additional 2.5 megabytes of disk space. It runs on Unix workstations (Sun and SGI) with X windows, and a Macintosh version is currently under development. The distribution of the Data Note System consists of a self-installing shell archive (shar file) which automatically configures the executables based on the environment in which it is unpacked. Thus, installing the system on a platform where Tcl/Tk is available requires typing one Unix command.

Acknowledgments. The author thanks Ellen Bergeman, Dan Davison, and Charles Lawrence for frequent discussions of the ideas in this paper. The research was supported by an appointment to the Human Genome Distinguished Postdoctoral Fellowships sponsored by the U.S. Department of Energy, Office of Health and Environmental Research, and Administered by the Oak Ridge Institute for Science and Education. A portion of the work was also supported by the W.M. Keck Center for Computational Biology and the Baylor Human Genome Center funded by the NIH National Center for Human Genome Research. The author is currently supported by DOE grant DE-FG03-94ER61818.

References

1. Chen IA, Markowitz VM. An overview of the Object Protocol Model (OPM) and OPM Data Management Tools, TR LBL-33706, Lawrence Berkeley Laboratory, 1994.
2. Durbin R, Thierry-Meig J. A C. elegans database. Available via anonymous ftp from lirmm.lirmm.fr, cele.mrc-lmb.cam.ac.uk and ncbi.nlm.nih.gov 1991.
3. Goodman N, Rozen S, and Stein L. LabBase: A Database to Manage Laboratory Data in a Large-Scale Genome-Mapping Project. *IEEE Engineering in Medicine and Biology.* Special issue on Managing Data for the Human Genome Project. 11(6) 1995.
4. Graves M, Bergeman ER, Lawrence CB. A graph conceptual model for developing human genome center databases. *Computers in Biology and Medicine.* Special issue on Information Retrieval and Genomics. 26(3), pp 183-197. 1996.
5. Mirkin BG, Rodin SN. *Graphs and Genes*, volume 11 of *Biomathematics*. Springer-Verlag, 1984

.

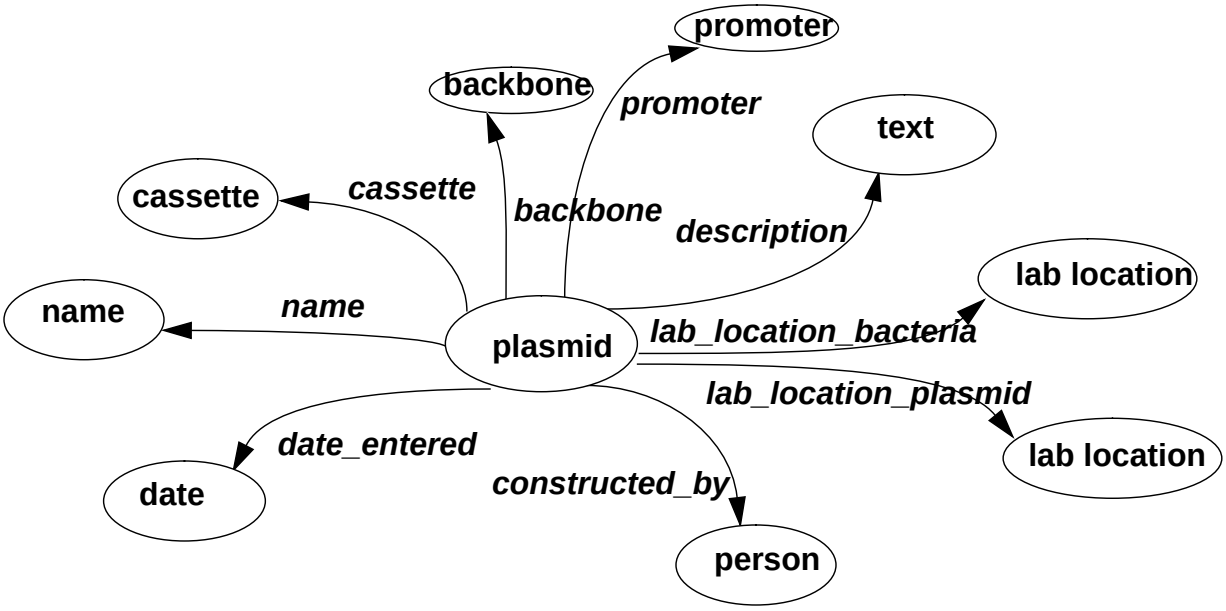

Figure 1: Schema of plasmid laboratory data

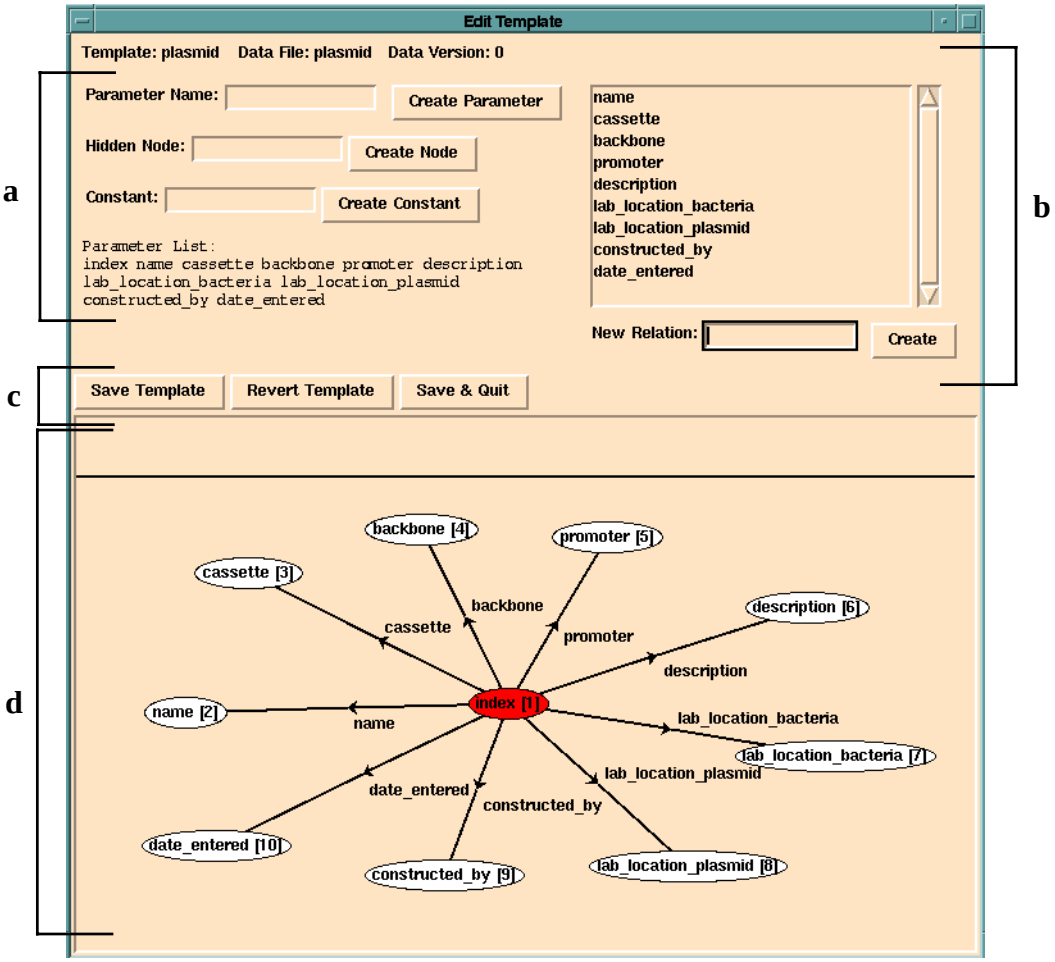

Figure 2: Edit template tool showing a graph template for plasmid laboratory data

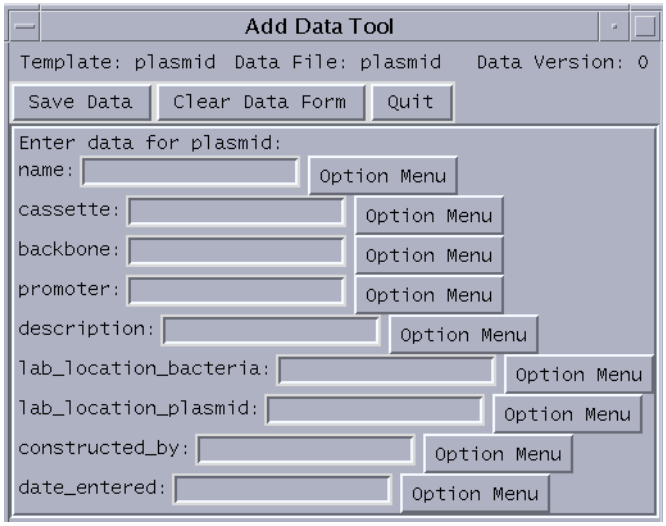

Figure 3: Data entry tool for plasmid laboratory data.

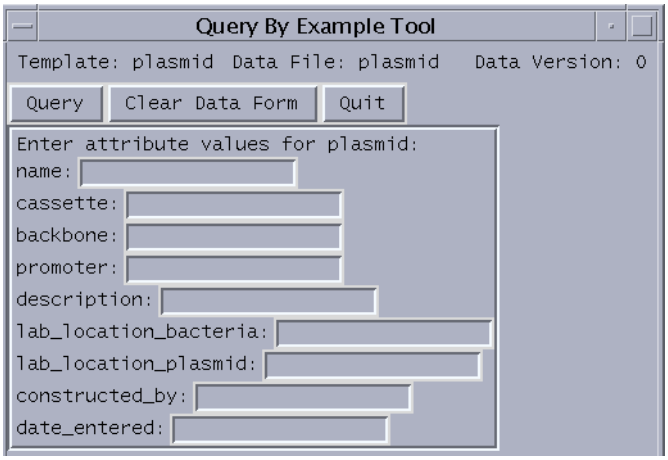

Figure 4: Query-by-example tool for querying plasmid laboratory data.

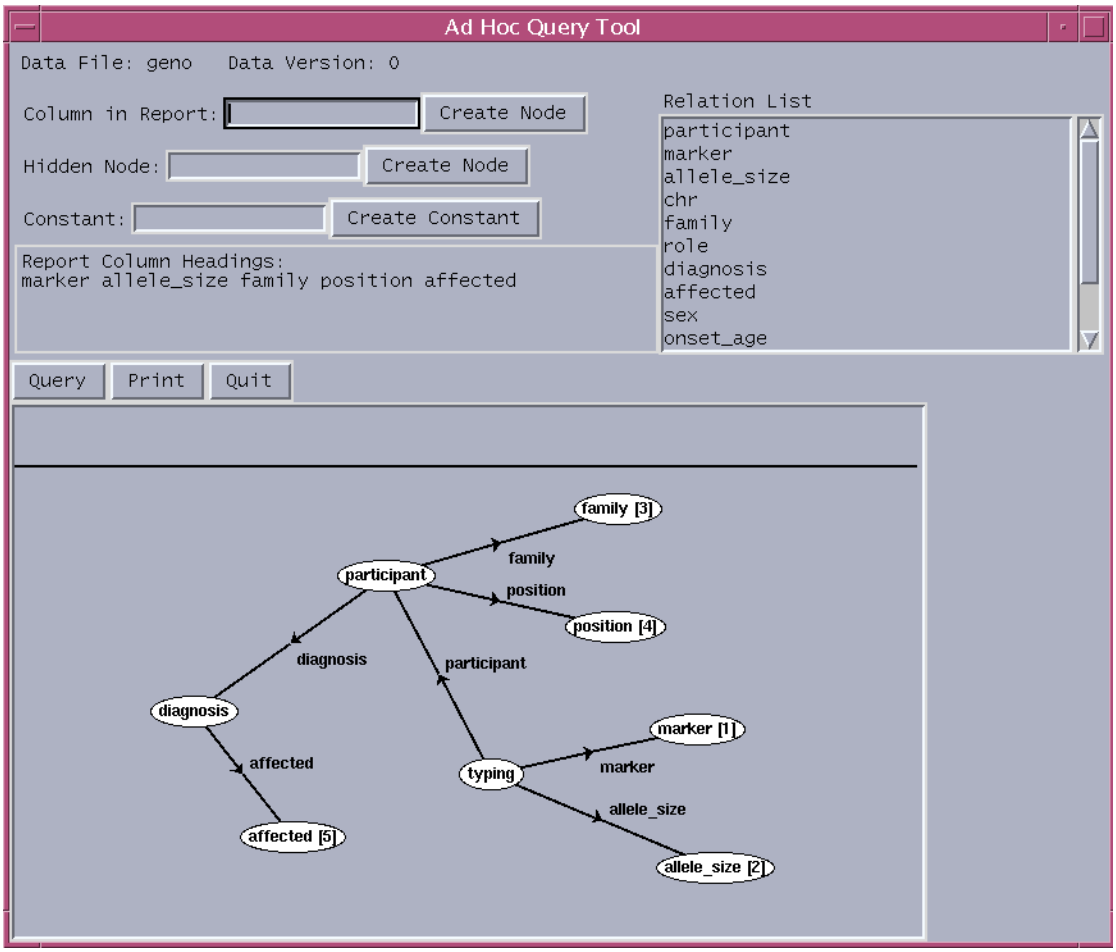

Figure 5: *Ad hoc* query tool showing a query of genotyping, diagnostic, and family data useful for genetic analysis.