Applied and Theoretical Electrophoresis (1995), **5**, 55–72

The Protein Disease Database of human body fluids: II. Computer methods and data issues

Peter F. Lemkin[1], Geoffrey A. Orr[2], Mark P. Goldstein[3], G. Joseph Creed[4], James E. Myrick[5] & Carl R. Merril[4]

[1]Image Processing Section/LMMB, Building 469, Room 150, NCI-FCRDC/NIH, Frederick, MD 21702; [2]PRI/Dyncorp, NCI-FCRDC, Frederick, MD; [3]Monoclonetics Int'l., Houston, TX; [4]LBG, NIMH/NIH Neuroscience Center, Washington, D.C.; [5]NCEH/CDCP Div. Environ. Health Lab. Sci., Atlanta, GA

The Protein Disease Database (PDD) is a relational database of proteins and diseases. With this database it is possible to screen for quantitative protein abnormalities associated with disease states. These quantitative relationships use data drawn from the peer-reviewed biomedical literature. Assays may also include those observed in high-resolution electrophoretic gels that offer the potential to quantitate many proteins in a single test as well as data gathered by enzymatic or immunologic assays.

We are using the Internet World Wide Web (WWW) and the Web browser paradigm as an access method for wide distribution and querying of the Protein Disease Database. The WWW hypertext transfer protocol and its Common Gateway Interface make it possible to build powerful graphical user interfaces that can support easy-to-use data retrieval using query specification forms or images. The details of these interactions are totally transparent to the users of these forms. Using a client-server SQL relational database, user query access, initial data entry and database maintenance are all performed over the Internet with a Web browser. We discuss the underlying design issues, mapping mechanisms and assumptions that we used in constructing the system, data entry, access to the database server, security, and synthesis of derived two-dimensional gel image maps and hypertext documents resulting from SQL database searches.

Keywords: Protein Disease Database, PDD, World-Wide-Web, hypertext, Internet, relational database, fold change, two-dimensional electrophoresis, acute phase proteins, databases, factual/standards, proteins/genetics, human, electrophoresis, gel, two-dimensional, blood proteins/analysis, CSF proteins/analysis, urinary proteins/analysis.

Introduction

The PDD is a database of proteins, diseases, and their quantitative relationships using data drawn from the peer-reviewed biomedical literature (Figure 1). The project is a collaboration among researchers at the NCI (design and software development), NIMH, and CDC (data collection, review and data entry). In our companion paper (Merril *et al.*, 1995), we discuss in more detail the biomedical issues and implications of having such a database. Here, we will briefly review some of the background and rationale for the database, and then present its initial implementation and future plans.

The examination of body fluids for disease markers dates back to antiquity. However, the study of protein abnormalities in specific body fluids such as serum was begun in earnest shortly after the turn of this century, with most of the quantitative observations reported in the last three decades (see review in Merril, 1995).

The difficulty is that while a considerable body of data about protein alterations in disease states has accumulated in the literature, it is fragmented, dispersed, and often difficult to access. In addition, reliability of the published data is of varying quality requiring proper statistical treatment for handling data from studies with different protocols. For these reasons and others discussed in the companion paper, we felt it was essential to develop this database.

Increasingly, research groups have been making their genomic and 2-D electrophoretic gel protein data available on the Internet.[1] Table 1 lists some World Wide Web 2-DE gel database servers. Many other genomic databases are already available on the Internet and discussions on how to access them are described in the special database issue of *Nucleic Acids Research*, September 22(17), 1994. Recent reviews of the 2-DE gel database literature (Hochstrasser, 1993; Celis, 1994), and a symposium (Pallini, 1994; Pennington, 1994) have identified many proteins involved in broad areas of medical interest. As will be discussed, we make use of these sources and other network databases by accessing them from our PDD server when necessary.

The Protein Disease Database (PDD) system

Observations on the limited availability of these quantitative data discussed in our companion paper (Merril,

Correspondence: P. F. Lemkin, Image Processing Section/LMMB, 469, Room 150, NCI-FCRDC/NIH, Frederick, MD 21702

Note: This work was first introduced at the Sept. 5–7, 1994 conference "2-D Electrophoresis: from Protein Maps to Genomes" in Siena, Italy—a working conference on two-dimensional electrophoresis and its link to genomes.

[1] The Internet is a global network of local and national networks linking universities, governments, industries, hospitals, and others, as well as individuals with a wide variety of services. A basic description of the Internet is given in (Krol, 1992).

Table 1 Some World Wide Web 2-D electrophoretic gel database servers. This list of 2-D electrophoresis WWW databases is being maintained by our Laboratory and is available in URL [http://www-ips.ncifcrf.gov/EP/EPemail.html]

- ExPASy (2-D liver, plasma, CSF, etc., SWISS-PROT, SWISS-2DPAGE, SWISS-3DIMAGE, BIOSCI, Melanie software)
  URL: http://expasy.hcuge.ch/
- CSH QUEST Protein Database Center (2-D yeast, REF52 rat, mouse embryo, Quest software),
  URL: http://siva.cshl.org/
- NCI Image Processing Section (GELLAB software),
  URL: http://www-ips.ncifcrf.gov/
- E.coli Gene-Protein Database Project—ECO2DBASE (in NCBI repository)
  URL: ftp://ncbi.nlm.nih.gov/repository/ECO2DBASE/
- Argonne Protein Mapping Group (mouse liver, human breast cell, etc.),
  URL: http://www.anl.gov/CMB/pmg_welcome-rev.html
- Cambridge 2-D PAGE (including beginnings of a rat neuronal database),
  URL: http://sunspot.bioc.cam.ac.uk/
- Harefield Human Heart 2-D gel Protein Database,
  URL: http://www.harefield.nthames.nhs.uk/
- Human Myocardial Two-Dimension Electrophoresis Protein Database,
  URL: http://www.chemie.fu-berlin.de/user/pleiss/dhzb.html

1995) stimulated the development of a protein disease relational database system that documents associations between patterns of protein concentration and disease states. Such a database allows finding the correlations of alterations in disease states to the quantitative changes in protein patterns in various protein assays including those observed in high-resolution electrophoretic gels. Note that we use the word "correlation" in the sense of association or relation—not necessarily in the sense of the statistical Pearson correlation coefficient.

Emphasis of the database is on proteins from human body fluids such as plasma, serum, CSF, and urine. These matrices were selected because of the ease of sample collection and data availability. The initial focus described in the companion paper is on the acute phase proteins (APP) (Mackiewicz, 1993). The purpose of collecting these data in the form of relative concentrations ("fold changes" defined below) is to allow investigators to explore relationships between disease states and protein patterns that are observed in a battery of protein assays. Improvements in 2-DE gel reproducibility makes it more feasible to compare 2-DE protein gels between laboratories and provides another method for putative protein identification.

While the initial focus is on the APP, emphasis is being made on collecting data on diseases of interest to the NIMH (neurological diseases), the NCI (cancer), and the CDC (toxicant exposures and genetic diseases).

Materials and methods

General

The characteristics and use of biomedical data used in the PDD help to determine the computer database requirements. We use the Internet with WWW and a Web browser to deliver this biomedical resource to researchers as a graphical data query and browser, and data entry tool. A key point of this paper is concerned with how we have integrated relational database management system (RDBMS) technology to support the PDD using these Internet tools. A RDBMS allows us to store relations between proteins, diseases, and references such that a broad range of qualitative and quantitative questions may be asked.

We will first describe the data and the paradigm for the database. Then we discuss the use of the World Wide Web on the Internet for delivering access of such an interactive database system to users. Figure 1 illustrates the general scheme for using quantitative fold change information derived from the literature.

The PDD is currently running on a dedicated SparcStation-2 computer with SOLARIS, that hosts both the WWW server and the RDBMS server used by the Working PDD. The Staging PDD server and database runs on a separate SparcStation-2 with SUNOS. Data are entered and checked on the Staging PDD and then manually copied to the working PDD. This will become more automated as we show later in the discussion. The initial database did not have much data as we were primarily concerned with eliminating bugs and developing a smoothly functioning data entry paradigm.

Biomedical research literature as a source for data

The driving notion behind the PDD is the observation that the peer-reviewed biomedical literature provides an overwhelming number of fragmented, unorganized, and disconnected findings. The volume of these findings can interfere with its usefulness. Because of this fragmentation, we are investigating the use of certified databases used in pathology and clinical laboratories for inclusion in the PDD.

While great progress has been made in bibliographic National Library of Medicine (MEDLINE and the Entrez subset) databases and highly focused single-entity databases (SWISS-PROT, GenBank, etc.), these sources, while useful, remain of limited utility in performing searches driven by quantitative relationships between entities, such as those between multiple protein concentration and/or activity profiles and disease states.

The use of such multiple entity quantitative research tools is possible only through the method of literature

evaluation and summarization that is closer to the area of meta-analysis (Mann, 1990; Peto, 1993; Powe, 1994) than to the traditional literature review. In meta-analysis, results from a number of studies (typically large-scale clinical trials) are combined to support statistical decisions taking into account the fact that the studies were probably performed with different protocols. The similarity of this method with meta-analysis is strong in the sense that both methods are retrospective, generally rely on "data mining" the published literature, and must deal with issues of publication bias, research quality, and comparability of data from different sources.

These methods differ in two important ways. While meta-analyses normally seek a conclusion, the approach taken in the PDD allows searches of the protein disease literature to suggest hypotheses by finding protein patterns associated with diseases. These patterns may then be further investigated using standard research methods. Further, while the statistical requirements of meta-analysis argue in favor of studying a well constrained set of therapies/diseases, the PDD approach encourages examination of relationships among sets of proteins and sets of diseases.

Normalization of data

Crucial to the success of this effort is a method of standardizing protein concentration and/or activity data across studies, since a wide variety of methods are used to measure and report them. Comparing data from different sources depends on normalizing them and then comparing the normalized values in searches. Our approach is to convert all results into a dimensionless quantity called *fold change*. We can then specify ranges of normal and abnormal fold changes. A protein's fold change is the ratio of its concentration in the two different states and is:

$$\text{fold change} = \frac{\text{mean of disease values}}{\text{mean of normal values}}$$

where the mean of a protein's concentration or activity is its mean as reported in the particular literature reference. The mean is really an estimate of central tendency such as arithmetic mean, median or mode. Any stochiometric protein assay method may be used for quantifying protein concentration including various immuno-assays, enzyme activities, colorimetric, 2-D gel quantitative analysis, etc. Therefore, the database is *not* limited in any way to data found in 2-D PAGE gels since any protein assay method may be used. However, proteins can often be located in 2-D PAGE gels in map images (cf. Figure 5) generated by the PDD and linked to other 2-DE gel and genomic databases.[2]

Qualitative changes are indicated by a fold change of 0 (if the protein is missing in the disease state) or infinity (if missing in the normal). An estimated value is just

[2] Even proteins not previously identified can generally be placed in an error box, denoting their theoretical location on a 2-D electrophoretogram, if their sequence is known (often predicted from the genomic DNA sequences).

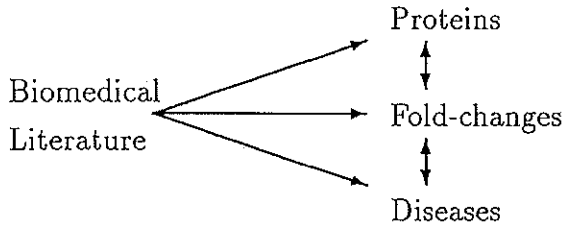

Figure 1 The Protein Disease Database paradigm. All data for the Protein Disease Database is derived from biomedical literature. Proteins are related to diseases by quantitative fold changes in their concentration or activity (*disease/normal*). All three of these entities are reported or derived from the literature. The fold change is the relation that ties proteins and diseases together in the relational database. The database can be accessed by stipulating a disease(s) or a protein(s) with a range of acceptable fold changes as part of the query.

that—a value estimated by the data in a particular study. There may be several different studies on the same diseases and proteins. We therefore need to search the PDD database on a range of fold-values where the normal and abnormal states have non-overlapping ranges. The fold change ranges may be calculated from the estimated value ranges that may be calculated directly in the PDD when adequate data is present. The worst case fold change ranges $[F_L{:}F_U]_N$ and $[F_L{:}F_U]_A$ can be estimated from the upper (U) and lower (L) bounds of the normal (N) and abnormal (A) values:

$$[F_L{:}F_U]_N = \left[\frac{A_U}{N_L}{:}\frac{A_U}{N_U}\right].$$

and

$$[F_L{:}F_U]_A = \left[\frac{A_L}{N_L}{:}\frac{A_U}{N_L}\right].$$

There are some other situations in which studies will contrast several different disease subgroups against each other rather than against the normal. In those cases, a different *fold change'* (i, j) could be calculated from the ratio of disease subgroups i and j. Disease subgroup attributes saved in the relational database could be used to identify that subset of data.

We recognize the statistical limitations of this method and strongly warn the user against reaching conclusions based solely on fold change data presented in the PDD because of differences in study design and protein detection methods (cf. Merril, 1995). Results from a PDD search should be treated as partially complete prescreening data which are subject to further analysis by the researcher to determine if they are relevant (much as is done in a Medline search).

Data entities used in constructing the PDD

The data consist of correspondences between proteins with diseases using fold changes as reported in the literature. Other quantitative features (to be discussed) are also captured in the data entry process and may be used in specifying searches.

For a literature reference to be used, it must be in refereed or certified literature and at a minimum present:

Table 2 Primary types of queries that could be answered with the PDD

- If proteins A, B, C, and D are increased, and proteins H and K are decreased, what disease entities might be present?
- For a given disease, specifically what pattern of fold changes might be expected for proteins A, D, and H?
- As a practical example: the patient has a lesion in the lung, but is it an infection or a tumor? What differences are seen in the APP patterns in each case—i.e., which sets of protein changes are important?

1) quantitative mean protein changes between normal and disease (or condition) states (including lower, upper, estimated, standard deviation) values, 2) a *p*-value associated with these changes, and 3) the number of patients (for both control and disease) used in the study.

Federated databases

Genomic databases are becoming available on the Internet as federated databases that can be thought of as collection and distribution services for these data (Mansfield, 1994). A federated database is a loose collection of databases that members of the federation and others can draw upon. Each specializes in a precisely defined domain. The advantages of sharing data are at least twofold: 1) databases do not have to be duplicated and maintained locally, thus greatly reducing the expense and time required, and 2) data are guaranteed to be the latest available for that database. Disadvantages are that: 1) there is a greater dependence on Internet connections being reliable, and 2) members of the federation share their resources with others. However, as computer systems get faster, these increased computational loads are less of a problem. In balance, the human workload is decreased and the power of the data increased by sharing federated data.

Data from some of these well maintained genomic databases are used to supplement PDD data when possible. If such federated data are available from remote WWW servers, then they are used internally by the PDD for various functions as well as being available to users as hypertext links. These include: 1) proteins-to-spots in viewable reference 2-DE gel maps—both in the PDD and in external databases, 2) proteins-to-identifiers in external databases available over the Internet such as the ExPASy system for SWISS-PROT (Bairoch, 1994) and SWISS-2-DPAGE (Appel, 1993; Appel, 1994), GDB℠ (Fasman, 1994), GenBank (Benson, 1994), and others, 3) synonyms of diseases or conditions linked to the Unified Medical Language System (UMLS℠) of the National Library of Medicine (UMLS, 1994), and 4) literature references to external MEDLINE℠ servers for recovering titles, authors, abstracts, and journal references. The use of a synonym database to resolve different terms will be critical for using the database for a wide variety of biomedical literature for both data capture and queries where terms for the same concept will have different names. The types of external genomic databases that have been discussed are examples of members of this loose federation of databases. The PDD uses this information and makes its disease correlation data available for other federation database servers to use.

Table 3 Examples of queries that could be answered with the PDD. A relational database offers flexibility in posing a wide variety of types of queries

- Diseases
  1. What diseases have fold changes with protein K?—changes of >5×?
  2. What diseases have fold increases >2.0× for protein A and B, and decreases <2.5× for protein C and D?
  3. Which diseases have a 0.5× fold change (i.e., decrease) in this protein?—in 'all' the proteins in the database?
  4. Compare protein patterns that change for diseases A and B.
- Proteins
  1. What proteins have fold changes associated with this disease? How much do they change? How much do they change in plasma? in urine?
  2. Which proteins change by more than 1.5× fold change (i.e., increase) in this disease?
  3. Where are these proteins in the serum (urine, CSF, etc.) reference 2-DE gel map? any 2-DE gel map?
  4. What is the name of this protein I am pointing to in the plasma 2-DE gel map? What diseases is it involved in?
  5. Show a 2-DE gel map of plasma proteins that increase in this disease in red and those that decrease in blue.
- Literature references
  1. What literature references discuss these proteins in the context of these diseases?
  2. What literature references for disease Q had more than N patients in the study?
  3. Who has worked on this protein?

Queries that may be answered by the PDD

Two basic types of queries addressed by this database are listed in Table 2. These involve searching for quantitative fold changes in protein concentrations between normal and disease states. Table 3 shows elaborations of these and other types of PDD-queries answered with this type of relational database.

The World Wide Web (WWW) paradigm—a review

A short review of the WWW paradigm may be of help in understanding how the PDD is designed and used (Figure 2). This section may be skipped for those who understand the WWW paradigm.

The WWW developed at CERN in Geneva, Switzerland is a network-based hypertext system for dynamically linking various information sources. The network in this case is the Internet that spans the world. An information source is any accessible body of multimedia files (text, image and sound) or databases that may be accessed from a "server" computer located on the Internet. Currently, most information sources are available free although there is some interest in eventually charging for high quality commercially valuable information sources. The WWW was initially conceived by Tim Berners-Lee to link high energy physics researchers, but its use has expanded ex-

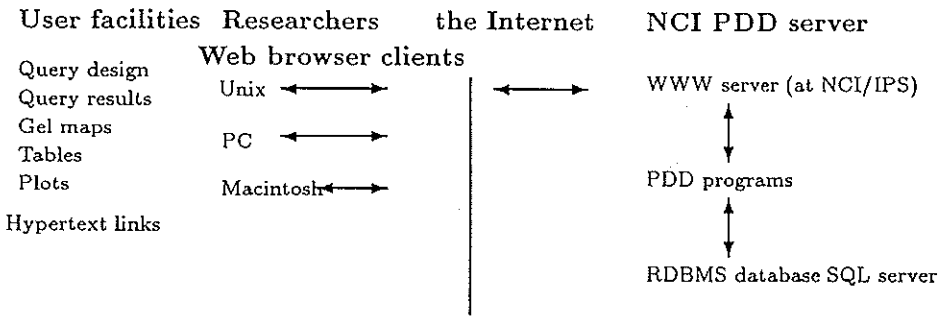

Figure 2 Relationship of Web browser, World Wide Web and the PDD. Researchers access the PDD database over the Internet through the WWW server at NCI by running the Web network browser. This then connects them with the NCI Image Processing Section WWW server to access the PDD. In response to user requests, the WWW server runs programs to query the RDBMS SQL server that in turn returns hypertext answers to user queries. These answers may contain hypertext links to other PDD data or to data in other WWW servers (such as SWISS-PROT or SWISS-2DPAGE, etc.). Web browser and the WWW server use the HTTP protocol. Documents with active (i.e., clickable) hypertext entries are generally written in HTML.

ponentially to include all types of information (Berners-Lee, 1994).[3]

The Mosaic Web browser program, (Andreessen, 1994; Schatz, 1994) was developed at the National Center for Supercomputing Applications at University of Illinois (NCSA). Another popular Web browser is Netscape by Netscape Communications Corporation. Whereas WWW is the collection of information sources distributed over the Internet, Web browsers are computer tools which run on your computer and allows you to browse this information (just as you might browse a book, journal or newspaper). Since there are a several WWW browsers available in addition to Mosaic and Netscape, in the remainder of the paper we will refer to browsers with similar capabilities as generic Web browsers. Browsers are available for UNIX, Macintosh and Windows-PC computers. Web browsers access WWW servers using the Internet TCP/IP protocol and a client-server message passing method. The latter use of TCP/IP implies that a full Internet connection is required. (See Dougherty, 1994 for a good introductory discussion on this.) Web browsers offer simple intuitive point and click graphical user interfaces to the WWW and the Internet. The user clicks on underlined text or active images called *hypertext references* and the Web browser responds by following the link to the associated information source and retrieving new information from WWW servers corresponding to those text or image objects. Associated with each hypertext reference is a Uniform Resource Locator (URL) that points to the exact information source at a particular Internet site. The Internet therefore may be thought of as a repository for distributed hypermedia. Figure 2 illustrates the relation of the Web browser to the PDD's WWW information source.

User Interaction using Web browsers: Users interact with Web browsers three ways: they 1) click on an underlined colored hypertext link to access a WWW server, which then provides them with the document pointed to by that link; or 2) click on an object in an active hypertext image (not all images are active); or 3) submit a form after filling in the various fields of the form. The action always causes the remote WWW server to return the requested information back to the Web browser.

There are several advantages in using a Web browser as a user interface. It is the first universal interface for the Internet that encourages worldwide collaboration. All information sources are handled transparently from the user's perspective as hypertext, which is all they usually see. The behavior of the Web browser program is the same regardless of whether it is running on a Windows PC, a Macintosh or UNIX workstation. Finally, all of these cross-platform Web browsers are available free from the Internet for those who want to download them using anonymous FTP (ftp.ncsa.uiuc.edu/Web).[4] The Netscape browser by Netscape Communications Corporation is available with anonymous FTP (ftp.mcom.com). Commercially supported versions of WWW browsers are also starting to become available.

Capabilities of the HTTP/HTML Paradigm used by the PDD: WWW servers use the HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) protocol to service requests by remote Web browsers. Requests may be for static documents (text, images, movies and audio clips), and dynamically created documents created by running programs (e.g., database search programs) on the server. HTTP documents are generally written in the HTML (**H**yper**T**ext **M**arkup **L**anguage). Both HTTP and HTML are discussed in Dougherty, 1994, but the best documentation is on the WWW itself, and is easily accessed using the built in help menus provided with most Web browser programs.

The HTTP and HTML paradigms support data entry forms, in-line and popup images in documents, and point-

[3] For more information on the The World Wide Web, see URL http://www.w3.org/hypertext/WWW/TheProject.html maintained by the W3 Consortium.

[4] The name ftp.ncsa.uiuc.edu is an Internet address of an "anonymous FTP" site. Anyone may attach to this site by running FTP on his/her computer and connecting to it. The login name is "anonymous" and the password is your e-mail address. Typing "Help" will list the commands and instruct you on how to use them. After attaching, move to the directory /Web. At this point you may look at the "directory" of files available and "get" files you are interested in, using "binary" transfer mode. Typically there is a README file that could be retrieved, read and then used to determine what files you should finally retrieve.

Table 4 Criteria for data quality of biomedical literature. These criteria are applied to biomedical literature that is to be included in the PDD. Additional data (such as external genome database identifiers and other study statistics) may also be included if available. These criteria are used by the PDD Editorial Board and Readers in validating data to be entered for data quality

1. It reports a controlled randomized clinical study.
2. It discusses a protein disease relationship in a clinical study describing quantitative protein changes (by some protein assay method) and the diseases or conditions involved.
3. The data are from a refereed journal article (books and certified databases etc. are being considered for use, but are not used now).
4. It includes concentration fold changes and values for disease and normal states.
5. It includes the number of patients in both the normal (control) and disease (case) populations used in a study.
6. A p-value for the difference in protein concentration between groups is given or can be calculated.

and-click interactive images as executable programs (called Common Gateway Interface or CGI programs). This paradigm enables us to implement fill-in form queries as well as data entry. The interactive images allow us to interrogate 2-DE gel maps by clicking on spots to access the database associated data. Finally, the CGI program extensions allow us to interact with the database in more interesting ways than just retrieving a document. In this context, Figures 2 and 8 show the collection of interdependent HTML documents, special HTML templates, and CGI programs using the RDBMS client-server SQL database used in the PDD.

Overview and architecture of the PDD system

We now continue our description of the PDD showing how it is used with the WWW. The PDD consists of a set of programs on the NCI/LMMB World Wide Web server. They use a locally developed SQL client/server relational database management system (RDBMS) to store data for the PDD, help build "smart" data entry and database query forms, and do query processing. Researchers use the Web browser program on their own computers to access the PDD Web server from the Internet.[5] The PDD translates simple user query requests entered into the Web browser to internal SQL query language statements for RDBMS client-server processing, and returns a hypertext document to the user in response. Figure 2 illustrates these relationships. The RDBMS and PDD programs are discussed in more detail in the Appendices.

Data capture and entry for the PDD: Data capture is a multistage process designed to ensure data quality. First, candidate data for the PDD are identified and recorded on paper forms by people whom we call Readers. Although completely electronic data entry is feasible, for reasons of facilitating data quality control we are currently using a mixed paper and electronic data capture paradigm. Initially, we will recruit readers as volunteers from the research community, later a more formal solution with paid readers might be considered. Readers are people competent to read and extract information from relevant journal papers in their biomedical areas or in other areas they have been assigned or offer to cover. The criteria used in evaluating whether a biomedical literature reference data would be used is listed in Table 4. We are also forming a data quality and relevance standards group for setting and enforcing this minimum criteria of data to be included in the PDD, such as is shown in Table 4.

The main steps in PDD-data entry and validation are:

1) Readers record this data on a standard paper form developed by the *PDD Editorial Board.* Use of a paper record helps enforce data quality because we can go back to the paper form at any point if necessary to verify data captured in the database. This of course will not eliminate errors made by the reader or in the articles themselves.
2) After the paper form is filled out by the Reader, the data are then entered into a Staging PDD database using the Web browser with PDD data entry forms.
3) Data are proofread and corrected by another individual using the PDD data browser in the Staging PDD.
4) The PDD Editorial Board members can review (i.e., browse) and validate new data in the Staging PDD by indicating that they are acceptable (or not) for migration to the publicly accessible Working PDD.
5) At this point, the new data set is allowed to be migrated to the Working PDD. The actual transfer will be done on a scheduled basis with checkpointing of the Working PDD before and after each update, to ensure the integrity of the database by allowing restoration of a previous version if required. Finally, periodic audits of the Working PDD for data quality are also planned. Such careful checking during data entry will help ensure data quality and robustness of the data in the database.

Initial data entry to the Staging RDBMS can only be performed by authorized personnel on remote Internet hosts using a distributed data entry paradigm. Note that these preliminary data are stored in the Staging database—not the Working PDD database. This allows several groups (e.g., NIMH in Washington DC, CDC in Atlanta, NCI/IPS in Frederick MD, and other sites) to participate in the data collection, review, and entry effort. All data captured and entered have timestamps, reader's, data entry clerk's, proofreader's, and reviewer's names automatically attached to RDBMS records for accounting purposes. Data entry using the `SimpleForm` program is discussed further in Appendix B. We can expect some backlogs of work to occur in some stages of the pipeline because of personnel bottlenecks. This problem might be handled by distributing the load to different individuals on the Internet since the work can be done from anywhere on the Internet.

Querying the PDD from Web browser using custom query forms: Queries are normally performed by filling out query forms in the Web browser and submitting them to the PDD. Users normally custom-design these query forms by selecting various processing and reporting options. Al-

[5] The PDD uses the `httpd` WWW server available from NCSA over the Internet (ftp.ncsa.uiuc.edu).

ternatively, PDD entries may be interrogated through interactive 2-DE gel maps by clicking the mouse button on protein spots. In the custom query form method, users may optionally select specific details (e.g., fold change range values, p-values, etc.) for each protein, disease or reference specified. The submitted query is mapped to SQL by the PDD program, which passes it on to the SQL RDBMS server. Search results are then reported back to the user as a Web browser document. This internal use of SQL in this process is totally transparent to the user. Although the underlying RDBMS uses the SQL query language, users do not need to know or use SQL.

Search results may be further analyzed and reported back to the user as lists of objects and their attributes, 2-DE gel maps, hypertext links to external federated WWW databases, and numerical tables or graphs derived from these data. These hypertext links to federated databases are generated by adding specific identifiers associated with the data in the PDD to the base Uniform Resource Locator. The federated database WWW servers then build a dynamic hypertext link to the PDD.

Summarizing the steps in this two stage process, users:

1. select a domain to search (proteins, diseases or references),
2. pick names of independent variables (proteins, diseases or references) used in restricting the search,
3. specify the general query constraints and how results should be shown,
4. submit the initial query to generate a specific query form.

Then, in the generated form, they:

1. supply additional specific values (such as ranges of fold changes, p-values, ranges of numbers of patients, or sample type) to the query form to further restrict the search,
2. select specific options for how the results are to be reported (tables, maps, graphs, or hypertext links),
3. submit the filled-in query form to the PDD to do the search.

Description of the PDD software

Figure 2 illustrates the top level structure and function of the PDD system. Aspects of the PDD software are explained in more detail in the Appendices. Some of this material assumes an understanding of the underlying methods used in WWW servers. Appendix A describes the RDBMS used by the PDD server programs when PDD data are required. Special Common Gateway Interface programs required to interface the WWW server with the RDBMS are described in Appendix B. Because the Web browser is used for all phases of the PDD, security is critical to protect the database. Appendix C describes some of the security issues and methods in the PDD necessary to protect the integrity of the database. Finally, to dynamically generate fill-in forms and process form options, a dynamic hypertext template language was developed. This is described in Appendix D.

Results and discussion

The PDD became operational in September 1994. As this paper is primarily about the computer database aspects, we refer you to our companion paper for more discussion on expected biomedical findings (Merril, 1995).

Since only the Web browser and an Internet connection are required to access the PDD, your direct on-line investigation of other aspects of the PDD system is encouraged. Figures 3–6 illustrate some of the aspects of the query forms as seen by users when requesting a search for diseases that change as a function of fold changes of a set of proteins.

Most bibliographic databases such as Medline are single-entity databases and are organized around proteins, sequences, genes, or diseases etc. Here, the data mined from the literature is organized around the concept of the quantitative relations of protein measurements to disease conditions.

The PDD represents a merging of several technologies that lets the user query for quantitative associations between protein patterns and diseases. The initial design of the PDD was specified as a stand-alone system for use on a workstation or high-end personal computer, and the database would have been distributed on CD-rom. However, as we observed the explosion of the WWW and the Web browser paradigm on the Internet, more interesting possibilities for a better distribution scheme became apparent.[6] Use of the Web browser as a client to the WWW server allowed us freedom from the problems of supporting different versions of software on different system platforms (Unix, MS Windows, Macintosh). Nor would we have to deal with CD-roms (with the many problems they entail including time, expense, distribution, inability to correct errors, and slow update cycles). We saw, along with many others, the utility of the WWW and Web browser client/server paradigm as a powerful data distribution medium for this type of biomedical data. We, therefore, redesigned the PDD to use the new Internet paradigm.

We selected the RDBMS data model for the PDD because the type of data we would use in protein disease correlations fits naturally into the relational model. Our decision to construct the initial underlying RDBMS SQL software engine in our laboratory enabled us to optimize it for the WWW environment (discussed in Appendix A), as well as to easily and inexpensively move it to more powerful UNIX multiprocessor platforms as computational demands require. We are in the process of converting the database engine to a commercial RDBMS to take advantage of a more robust environment.

Problems encountered with the WWW/Web browser "stateless" paradigm

Because we had decided to use the WWW/Web browser Common Gateway Interface (CGI) paradigm, we had to

[6] The number of Internet hosts from 1974–84 was 0 to 1,000; from 1984–92 was 1,000 to 1,000,000; and from 1992–2000 is expected to reach 100,000,000 as estimated by the Internet Society.

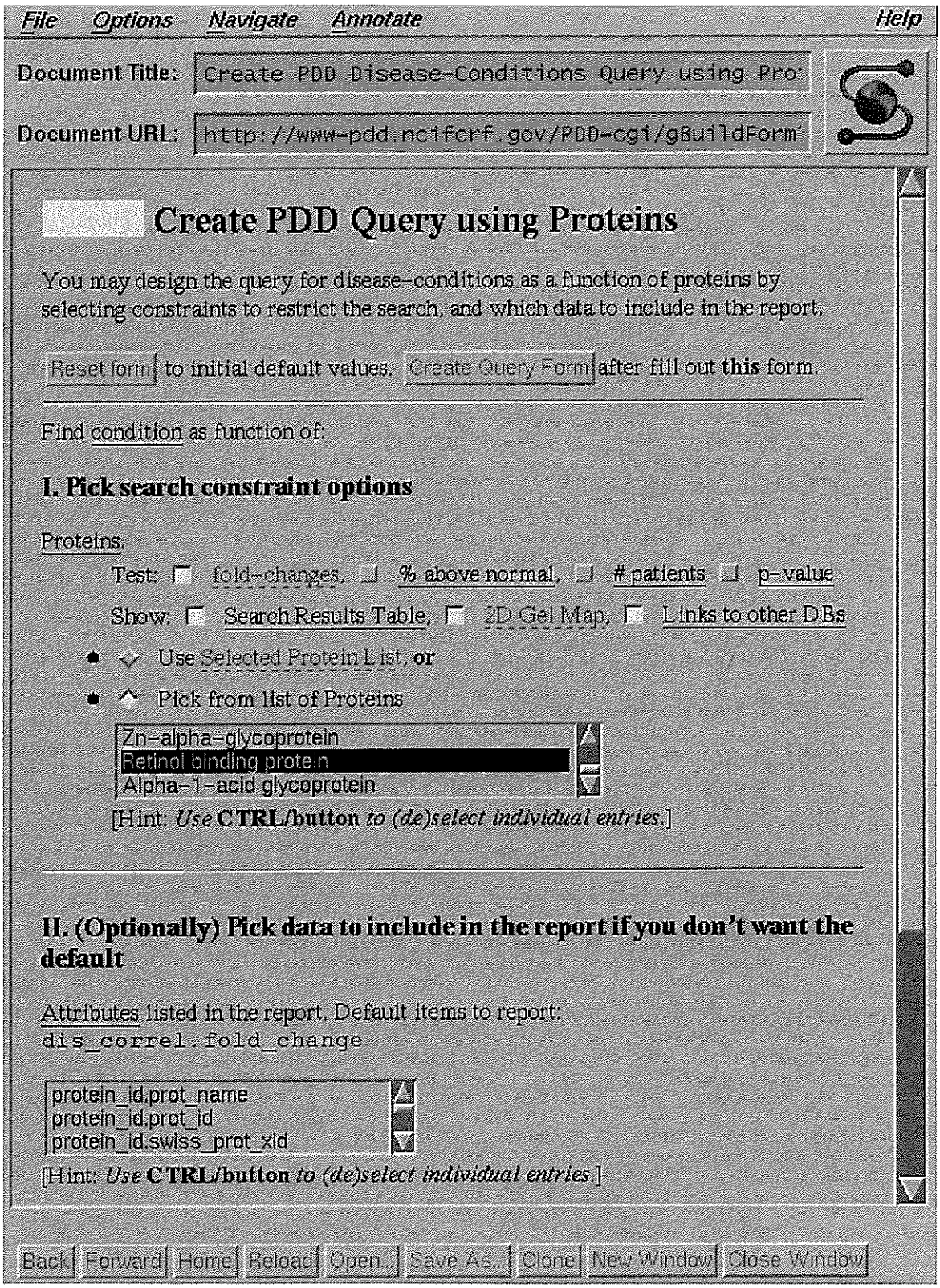

Figure 3 Example of an initial Query form. This illustrates the first stage query where we design the second stage query that will "find" instances of proteins, diseases or references as a "function of" proteins, diseases or references. Of course the "find" and "function-of" domains are not the same. This example finds disease conditions as a function of specified proteins. Here, the user selects a set of proteins to be used in the search, the quantitative tests to be performed on these proteins, and how the results should be presented. Submitting this form results in a more detailed query shown in Figure 4.

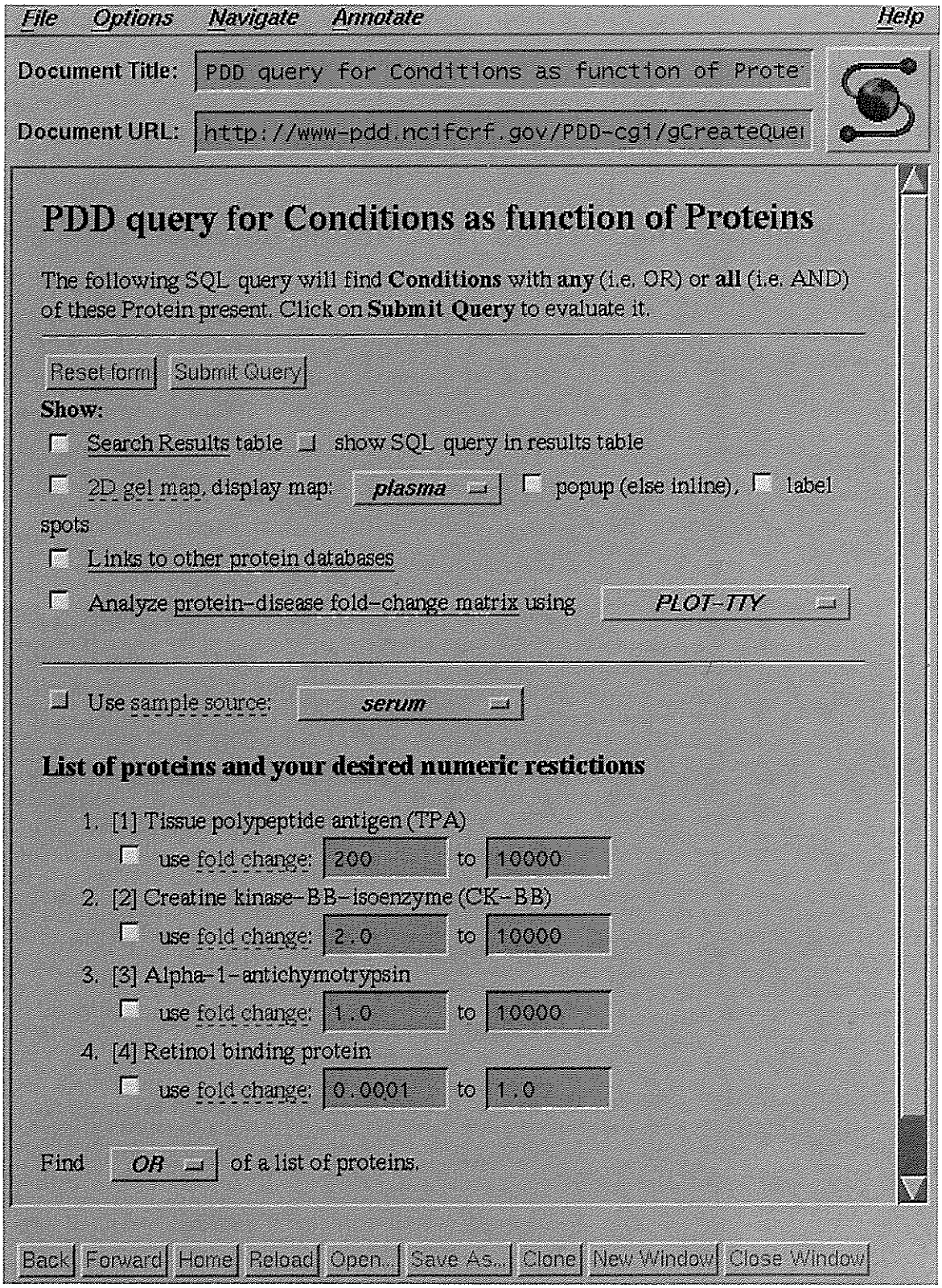

Figure 4 Example of an secondary Query form. This illustrates a detailed query to find disease conditions as a function of specified proteins. Here, the user specifies the limits on fold change for the proteins they are querying before doing the search. Submitting this form performs the search with results being returned to the user, of which some are shown in Figures 5 and 6.

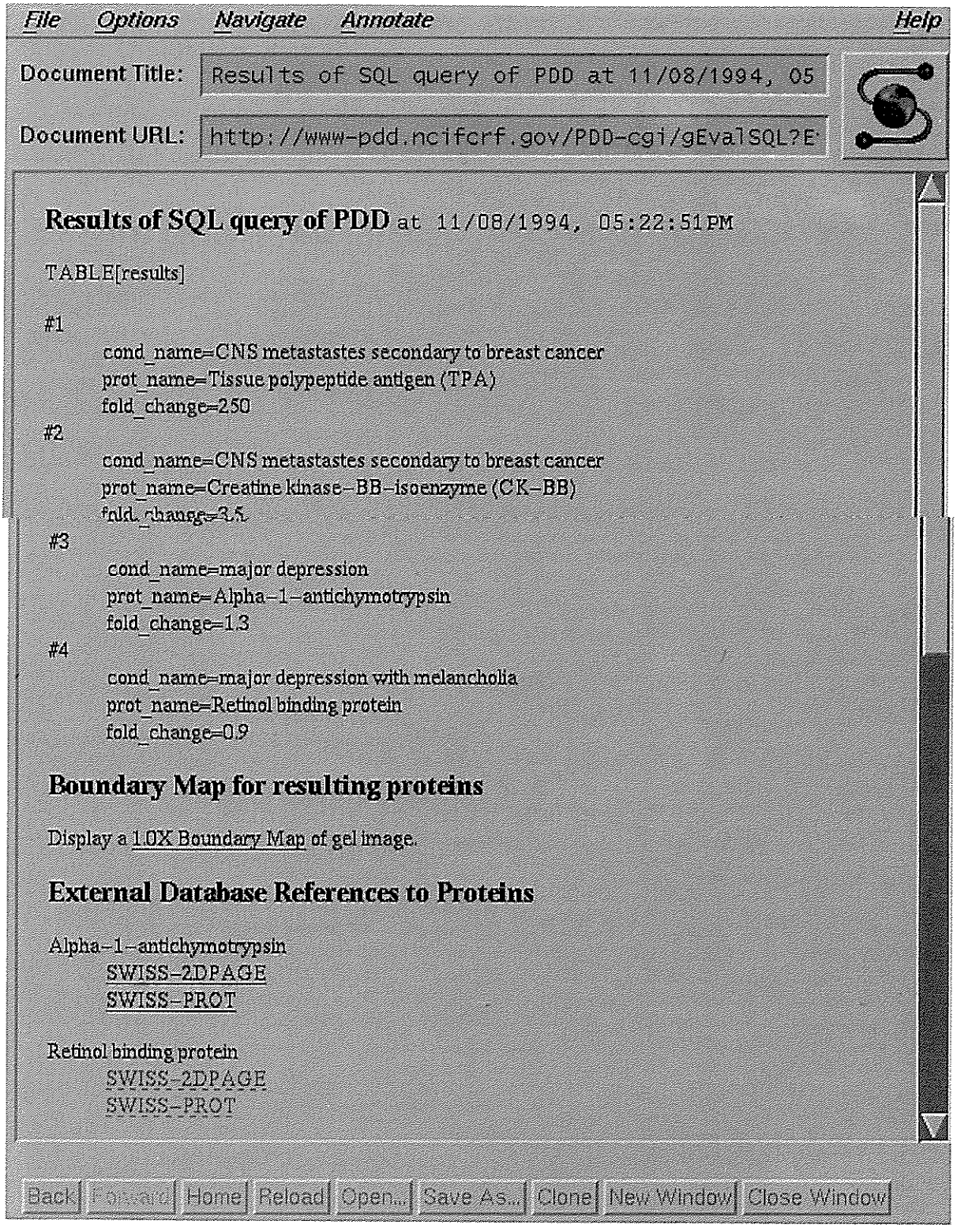

Figure 5 Example of some tables and hypertext links in search results from a query. This illustrates some of the results of finding disease conditions as a function of specified proteins based on the conditions specified in Figure 4. Additional results (not shown) can be tables and plots that summarize the data along with various statistical correlations and measures of protein fold expression profile similarities presented in both tabular and graphical form.

work with the constraints that it imposed. Some of the constraints include being "stateless", being transaction oriented, and lacking security. We subsequently developed a set of CGI programs that run on the NCI WWW server as requested by Web browser users. Since the WWW HTTP protocol is stateless (i.e., each hypertext "click" starts a request that ends with a response from the WWW server with no information saved on the server), we need-

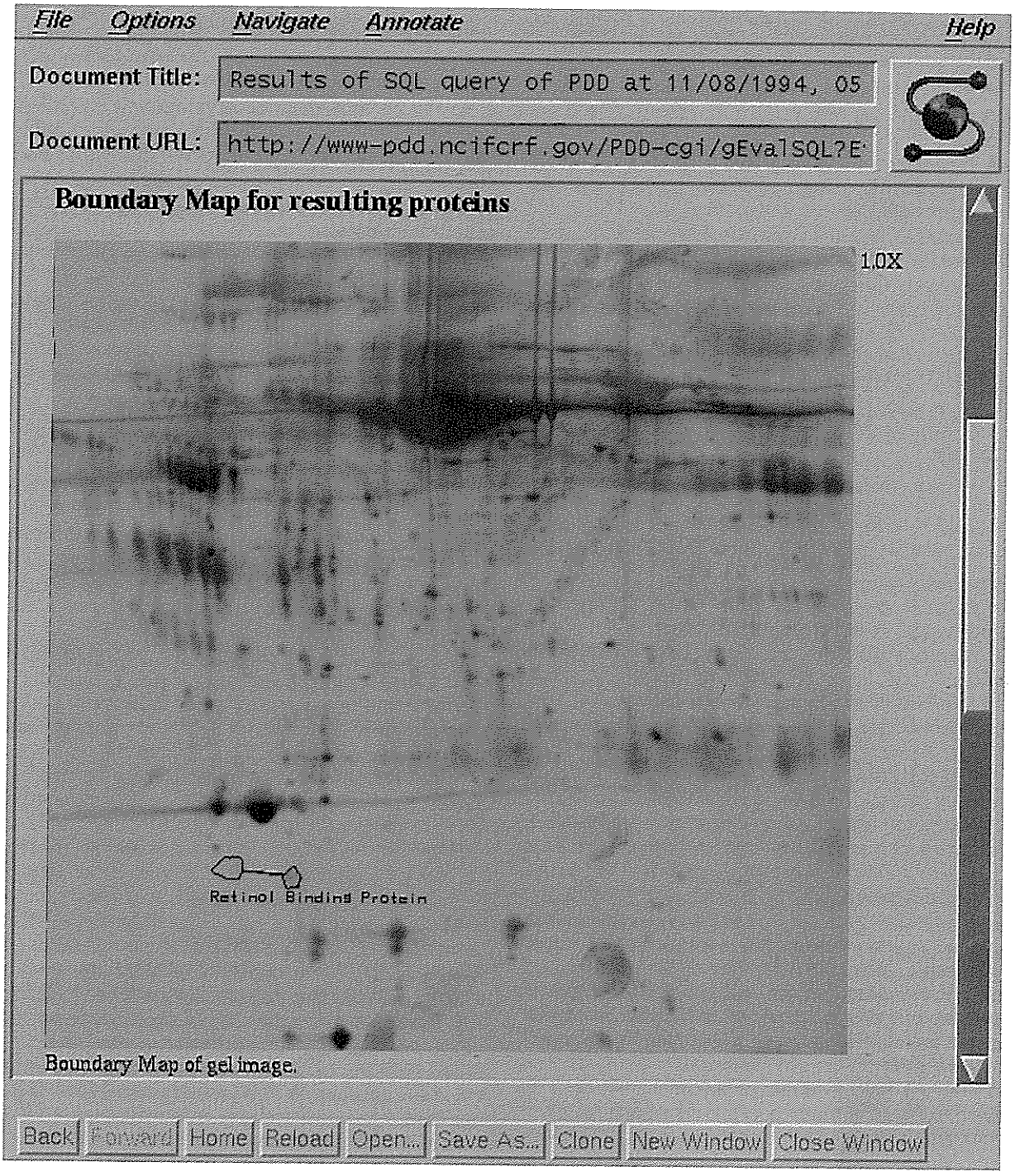

Figure 6 Example of a 2-DE Gel Map in search results from a query. This illustrates some of the results of finding disease conditions as a function of specified proteins based on the conditions specified in Figure 4. Proteins that increase (decrease) in all diseases are drawn in red (blue). Proteins that increase in some and decrease in others are drawn in yellow.

ed a way to track incremental work associated with each user. The need for this can be seen for query evaluation where we may define or find lists of several different types of objects (proteins, diseases, or references). These may be used at some later point in a session to help frame other queries. The problem of adding "state" information was solved by embedding "hidden" data in the HTML forms that are generated by the PDD and on which the user will make further selections before submitting them. The hidden state data are then passed back to the PDD. So, in effect, state information lives in the user's Web browser current document.

Other related issues arose. First, since all access to the database is from the Web browser for both users and data entry personnel, we had to implement security measures to protect the integrity of the PDD database from accidental or malicious use. Security is required at several levels and is discussed in Appendix C. These security

measures limit user access to the data entry facility as well as prevent them from changing the hidden SQL in the generated forms from their Web browser window. Second, because of the dynamic nature of queries, we had to dynamically generate query and other forms based on current content of the RDBMS. This was solved by developing a HTML template language used for expanding templates into HTML using data from the RDBMS as required (as discussed in Appendix D).

Because the WWW HTTP protocol is stateless, it is generally more difficult to build a graphical user interface that is as intuitively satisfying as can be done if the system were constructed entirely under X-Windows or Microsoft Windows. Therefore, solutions were implemented for the PDD that work, but are not as interactive as we would like. As the HTTP/HTML protocols improve and these problems are addressed by the general WWW community, we will improve our interface to take advantage of these advances.

Limitations on how the database is to be used

Users are reminded that the *p*-value normally associated with a hypothesis test is the probability that the observed difference is because of chance, and that the accuracy of the *p*-value is subject to errors in both experimental and statistical methods (in both conception and execution). Many studies that have reported impressively small *p*-values have supported conclusions later shown to be erroneous. Replication is the strongest form of scientific confirmation.

Use of concentration or activity fold change is a convenient means of making protein measurement unit differences transparent and allowing comparability between studies, but this comparability can be misleading. This "normalization" method uses only an estimate of central tendency or (mean, median, or mode) to represent the distribution of data, and ignores the role of dispersion (sample variance and other distributional characteristics such as skewness and kurtosis). The PDD does record sample ranges, estimated values, and standard deviations when reported, but this information is frequently not included in the articles. Because of these factors, comparisons between studies cannot be reliably made at the quantitative level unless details of the protocols used in the studies described in the papers are taken into account by the PDD user. For these reasons the user should always consult the original papers presented by the PDD query result.

It is also true that the database will likely be biased in favor of positive results, since negative findings tend to be undereported (Szklo, 1991; Dickerson, 1993; DuRant, 1994). The user of the PDD must consider all these factors, and bear in mind that use of the PDD may result in erroneous conclusions unless the data and experimental details provided in the reference publications are critically analysed.

The goal of the PDD is to provide a powerful search tool for the protein disease biomedical literature, and to allow rapid identification of patterns of protein disease correlations. The degree of sophistication of the search is dependent on the detailed level of quantitative data collected and the software provided. The user must also be reminded that powerful tools do not automatically provide accurate results.

At the current stage of development, use of the PDD should be limited to exploratory use, i.e., rapid and flexible identification of relationships that should be further studied through direct review of the literature, and most certainly through standard laboratory and clinical research techniques.

We believe that this database is an early precursor of systems that will ultimately be used to help guide the diagnosis and treatment process through identification of relevant literature. Despite the limitations discussed above, the PDD provides a useful tool to the researcher of today, and a versatile testbed for more powerful systems of the future.

Limitations of the database

The initial database is still small as we are finalizing the schemas for data, and the data acquisition process. We are now embarking on a data capture and entry effort and expect the amount of data in the database to greatly increase during this year as additional groups have volunteered to become readers.

Although we have initially concentrated on the APP class of body fluid proteins, we don't limit the PDD to the APP and are pursuing other families of proteins. We do not require that proteins be found or identified in 2-DE gels to be used in the PDD, only that they may be assayed in human body fluids and that there be useful protein disease correlations based on a quality biomedical literature source.

It should be pointed out that although most proteins are not identified in 2-D PAGE gels, increasing numbers of proteins are being identified by various methods including microsequencing, immunoblotting, amino acid composition, mass spectrometry, and other methods (Celis, 1992). These identified proteins are being included in Internet-accessible databases such as ExPASy (Appel, 1994) and others listed in Table 1. Because more and more 2-DE gel groups are running similar IPG gel protocols (Chiari, 1992) and incorporating better cross-linkers in the gel chemistry (Hochstrasser, 1988), it is becoming easier to compare gels of similar material between groups to identify many of the proteins.

Database quality issues for data entry

A key advantage of using the World Wide Web is that users do not have to be computer experts to use the system. However, creation of a database of any kind requires that close attention be paid to establishment and maintenance of data quality. We have devised an initial protocol to help enforce data quality. We envision modifying the protocol as required as we gain more experience with the process.

In the PDD, data proceeds from the Readers (who collect and record data), to Data Entry personnel, to Proof-

readers, to the PDD Editorial Board, and finally to the PDD that is accessed by end users.

The fundamental philosophy here is that no database is completely error-free, and that while aggressive steps must be taken to prevent data errors from occurring, a full program designed to detect and correct errors must also be carried out. So, we are using two separate databases: the Staging database, where data are entered and verified, and the Working database, which provides data to the PDD system as seen by users. Data are transferred from the Staging to the Working database only after three key elements of data quality have been examined and verified.

The first of these, data entry accuracy, is established and maintained by performing a 100% verification of all data entry before its transfer to the Working database. This verification will be performed by Proof-readers who compare each record entered with the paper form from which it was transcribed. These Proof-readers, who are trained to access the Staging database through their Web browser, could be office staff associated with the Readers.

Next, interpretation and data recording quality are maintained by the Editorial Board and by Readers, who will conduct periodic quality audits of data held as paper form records. These audits involve comparison of the paper form with the source article by someone other than the original Reader, and will require resolution of inconsistencies or ambiguity of interpretation.

Finally, research article quality is maintained by the PDD Editorial Board, which will publish and enforce guidelines and a checklist of inclusion/exclusion criteria for papers used in the database. Compliance with these standards is verified before data from a paper are placed in the Working database.

In each of these cases, deviations from established criteria and inconsistencies between article, paper form, or computer record will be reported to a central data manager/database administrator, who coordinates resolution of the error, and correction of the database. The challenge is to create a distributed data entry system that preserves data quality while not also at the same time creating a bureaucracy. We have and will be developing computer tools to help support this paradigm.

Future directions for expansion of the database

The initial design of the database has changed several times during its development and is now relatively stable. This does not mean however, that the database design, statistical analysis, graphical presentation methods, and type of data are frozen. On the contrary, we are receptive to adding other mined literature data that the scientific community feels could be useful in relational searches. The analysis paradigm is being expanded to do more complex searches as well as let us better handle disease subgroups using *fold change'*(i, j) data described in the Materials and Methods. More interaction with other federated databases is being added to make all aspects of the system more robust and easier to use. We will be providing access of individual PDD correlations to other federated databases by hypertext links using the PDD correlation identifiers.

Other aspects of this data acquisition model will be reported in future papers. Also being refined are mechanisms for comparing 2-DE gel map images on the PDD database (or other Web servers) with user gels on the user's own computers. This makes sense if user gels were produced with similar protocols. It would help users locate putative protein identifications in their own gels, suggesting targeted experiments to confirm these identifications.

Although the initial RDBMS was adequate for testing the feasibility of the system, we are migrating the RDBMS to a commercial system such as Oracle. Although we expect this to be somewhat slower, it should be more robust and easier to maintain.

Availability of the Protein Disease Database

The PDD is still undergoing changes and data are now being added to the database. We will be allowing limited access to the server when possible during this initial period. Users are encouraged to register with the PDD server to create a PDD account since it will then enable them to keep private "selected object lists" associated with their user account. These lists should remain on the system for a day or more (resources permitting).

The Uniform Resource Locator (URL) used by Web browser to access to the PDD is:

`http://www-pdd.ncifcrf.gov/.`

When optional registration is used, a user name and password are returned to the user for future accessing the PDD. If they choose not to register, they would just enter the system by clicking on the "Access Protein Disease Database" button and the PDD will assign the user name "demo" with password "demo" to them for the duration of the session.

We solicit comments from users on errors, new data or types of data desired, and other suggestions. We may be contacted by E-mail at `pdd@ncifcrf.gov`.

Acknowledgements

The PDD has been a team effort influenced by helpful suggestions from many sources. In particular we thank Edward Whitley for his help in developing the early database concepts and entering some of the early data. Ann Barber, Jacob Maizel and Greg Alvord had useful suggestions for this manuscript. We also thank our colleagues and the many people around the world who have dropped in on the PDD from the Internet from time to time and have given us their feedback. Of course, we acknowledge the CERN group for their developing the World Wide Web concept and the NCSA group for developing the Mosaic browser that has opened up so many avenues for globally sharing data.

References

Andreessen, M. & Bina, E. (1994). Mosaic. *Internet Res. Electron. Networking Appl.* **4**(1), 7–17.

Appel, R.D., Sanchez, J.-C., Bairoch, A., Golaz, O., Miu, M.,

Vargas, J.R. & Hochstrasser, D.F. (1993). SWISS-2DPAGE: a database of two-dimensional gel electrophoresis images. *Electrophoresis* **14**(11), 1232–1238.
Appel, R.D., Sanchez, J.-C., Bairoch, A., Golaz, O., Rivier, F., Pasquali, C., Hughes, G.J. & Hochstrasser, D.F. (1994). SWISS-2DPAGE database of two-dimensional polyacrylamide gel electrophoresis. *Nucleic Acids Res.* **22**(17), 3581–3582.
Bairoch, A. & Boeckmann, B. (1994). The SWISS-PROT protein sequence data bank. *Nucleic Acids Res.* **22**(17), 3578–3580.
Benson, D.A., Boguski, M., Lipman, D.J. & Ostell, J. (1994). GenBank. *Nucleic Acids Res.* **22**(17), 3441–3444.
Berners-Lee, T.J., Cailliau, R., Luotonen, A., Henrick, F. & Secret, A. (1994). World Wide Web. *Comm. Assoc. Comp. Mach.* **37**(8), 76–82.
Celis, J.E. (Ed). (1992). Special issue: *Two-dimensional Gel Protein Databases. Electrophoresis* **13**, 891–1062.
Celis, J.E. (Ed). (1994). Special issue: *Electrophoresis in Cancer Research. Electrophoresis* **15**, 305–556.
Chiari, M. & Righetti, P.G. (1992). The immobiline family: "vacuum" to "plenum" chemistry. *Electrophoresis* **13**, 187–191.
Dickerson, K. & Min, Y.I. (1993). Publication bias: the problem that won't go away. *Ann. NY Acad. Sci.* **703**, 135–148.
Dougherty, D., Koman, R. & Ferguson, P. (1994). *The Mosaic Handbook for the X Window System,* O'Reilly & Associates, Sabastopol, CA. ISBN-1-56592-094-5; . . . *for Microsoft Windows,* ISBN-1-56592-095-3; . . . *for the Macintosh,* ISBN-1-56592-096-1.
DuRant, R.H. (1994). Checklist for the evaluation of research articles. *J. Adolescent Health* **15**, 4–8.
Elmasri, R. & Navathe, S.B. (1994). *Fundamentals of Database Systems,* 2nd ed. Benjamin/Cummings Pub.: NY.
Fasman, K.H., Cuticchia, A.J. & Kingsbury, D.T. (1994). The GDB Human Genome Data Base anno 1994. *Nucleic Acids Res.* **22**(17), 3462–3469.
Hochstrasser, D., Harrington, M.G., Hochstrasser, A.C., Miller, M.J. & Merril, C.R. (1988). Methods for increasing the resolution of two-dimensional protein electrophoresis. *Anal. Biochem.* **173**, 424–435.
Hochstrasser, D. & Tissot, J. (1993). Clinical Applications of two-dimensional gel electrophoresis. In *Advances in Electrophoresis–Vol 6,* A. Chrambach, M.J. Dunn, B.J. Radola (Eds), VCH Pub.: NY, pp. 267–375.
Krol, E. (1992). *The Whole INTERNET User's Guide & Catalog.* O'Reilly & Associates, Sabastopol, CA. ISBN-1-56592-025-2.
Mackiewicz, A., Kushner, I. & Baumann, H. (1993). *Acute Phase Proteins—Molecular Biology, Biochemistry, and Clinical Applications.* CRC Press: Boca Raton, Florida.
Mann, C. (1990). Meta-analysis in the breach. *Science* **249**, 476–480.
Mansfield, B.K. (Ed). (1994). Future plans for databases. *Human Genome News* **6**(3), 4.
McCray, A.T. & Razi, A. (1995). The UMLS Knowledge Source Server. To appear in: *Proceedings of MEDINFO '95,* Vancouver, B.C., Canada, July 23–25, 1995. [Info on UMLS may be obtained from wth@nlm.nih.gov].
Merril, C., Goldstein, M., Myrick, J., Creed, J. & Lemkin, P.F. (1995). The Protein Disease Database of human body fluids: I. Information management needs and fundamental design considerations. *Applied Theoretical Electrophoresis.*
Pallini, V., Bini, L. & Hochstrasser, D. (1994). *Proceedings: 2D electrophoresis: from protein maps to genomes.* Univ. of Siena, Italy, Sept 5–7, 1994.
Pennington, S. (1994). 2-D protein gel electrophoresis: an old method with future potential. *Trends Cell Biol.* **4**, 439–441.
Peto, R., Collins, R. & Gray, R. (1993). Large scale randomized evidence: large, simple trials and overviews of trials. *Ann. NY Acad. Sci.* **703**, 314–340.
Powe, N.R., Turner, J.A., Maklan, C.W. and Ersek, M. (1994). Alternative methods for formal literature review and meta-analysis in AHCPR patient outcomes research teams. *Medical Care* **32**(7), JS22–JS37.
Schatz, B.R. and Hardin, J.B. (1994). NCSA Mosaic and the World Wide Web: global hypermedia protocols for the Internet. *Science* **265**, 895–901.
Szklo, M. (1991). Issues in publication and interpretation of research findings. *J. Clin. Epidemiol.* **44**, Suppl. I, 109S–113S.

Appendices

A. Design of NCI/IPS relational database system

The PDD database software was constructed using a SQL client/server Relational Data Base Management System (RDBMS) to store data and do query processing. This RDBMS software engine was constructed at the Image Processing Section (IPS) of the NCI to optimize its use for the PDD. Much of the design was based on algorithms described in (Elmasri, 1994). Details of this RDBMS software will be described in later papers. Some of the key features are:

1. memory-based relational tables for speed and simplicity rather than disk-cache based. Data tables are loaded into memory from disk files when it starts up the database, and checkpointed (written) back to disk files when SQL UPDATES are performed,
2. a grammar based parser that supports an increasingly complete SQL subset,
3. query optimization support in query evaluation routines,
4. a multi-threaded architecture that can take advantage of multiprocessor computers for increased throughput,
5. SQL client/server access uses a TCP/IP socket interface with a simple message based protocol protected by time-stamped encryption,
6. special HTML relational table output formatting support for WWW servers,
7. runs on SUNOS and SOLARIS on SUN UNIX workstations, and
8. POSIX standard C as implementation language allowing porting to other hardware and operating systems.

This RDBMS design is modular and therefore easy to modify to support object-oriented operations for enhancing the PDD. Being memory based, it is also faster than disk-based cached RDBMS (although the latter in the form of commercial systems such as Oracle or Sybase, or the Berkeley Postgres client-server systems could be substituted). We are investigating migrating the RDBMS to a commercial system such as Oracle. Figure 7 shows a simplified version of the RDBMS entities used to describe the PDD data. The full relational schema may be accessed from the PDD server itself using the Web browser interface.

Table	Attributes
au_list	(author names and author id)
au_ref	(reference id and author id)
full_ref	(reference id and description)
protein_id	(protein name, protein id, external references)
source	(sample source name and source id)
prot_assay	(assay method name and assay id)
assay_unit	(assay unit name and assay unit id)
condition	(disease name and disease id)
dis_correl	(disease,protein, source, assay, assay unit, and reference id numbers, fold change data, sample sizes data, p-value and comments)

Figure 7. Entities of RDBMS Schema of PDD. This shows some of the tables and entities stored in the Protein Disease Database. An id is generally a number, but any unique symbol will do. Not shown is the edit date and editor associated with each record for accounting purposes. The full relational schema is not shown here, but is available in the PDD.

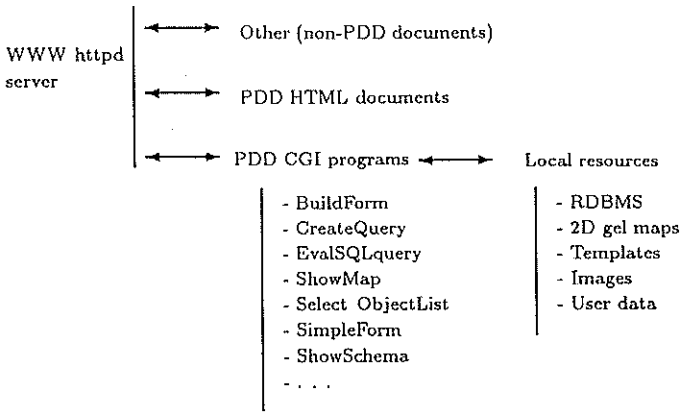

Figure 8. Organization of the PDD httpd WWW server. The WWW server invokes PDD documents and Common Graphics Interface programs. These in turn use local data resources including template files, 2-DE gel images and maps, as well as the relational database management system RDBMS where PDD data are stored and queried.

B. Connecting WWW with PDD's RDBMS using CGI programs

Some PDD Common Gateway Interface programs dynamically create HTML documents to pass back to the Web browser user. They do this by getting user input parameters specified in HTML forms from the Web browser as well as from the user's local state file kept on the PDD server. The RDBMS is called as required to expand user requests (for example to include a list of proteins or diseases as a "pick list" menu) in the new document returned to the user's Web browser window. The HTML data entry `FORM` protocol allows: text input (with and without scrolling), check boxes, radio buttons, pull-down menus, multiple-object selection windows, etc. This `FORM` protocol data is then analyzed by the CGI programs invoked by the WWW server. Some of the CGI programs used in supporting the PDD include:

1. `BuildForm`—translate special templates into HTML.
2. `CreateQuery`—generate query forms from user specification.
3. `EvalSQLquery`—evaluate user query by calling RDBMS server.
4. `SelectObject`—edit selected Protein, Condition or Reference object lists (to be discussed).
5. `ShowMap`—generate 2-DE gel map images of selected proteins.
6. `SimpleForm`—map data-entry form data to SQL updates for RDBMS.

The `BuildForm` CGI programs maps special PDD template files kept in the PDD server into HTML for display by the user's Web browser window. Template processing involves several operations such as conditional and macro symbol expansion, "include" files, and calling the RDBMS to get the most recent lists of objects (proteins, diseases, references, etc.) for inclusion in synthesized Web browser forms, menus, or review-lists of objects. Appendix D discusses HTML template processing in more detail.

Query processing is a two stage process. The first stage selects objects to be used in the query (e.g., picking a specific list of proteins, as well as options specifying how to compute the search and how to show the results). The `CreateQuery` CGI then dynamically synthesizes specific query forms using these parameters to generate the second stage query. Figure 3 shows a first stage query form. Figure 4 shows the resulting second stage query form that was generated.

Submitting the second stage query invokes the `EvalSQLquery` CGI program that expands and evaluates the resulting SQL query statements by:

1. expanding instantiated template form-variables and conditional Boolean subexpressions in the SQL query,
2. sending this to SQL server for evaluation by the RDBMS,
3. mapping search results (relational tables) to HTML, hypertext links, graphics plots or 2-DE gel maps,
4. returning mapped search results for display by Web browser.

The PDD then generates 2-DE gel maps and hypertext links to other WWW servers such as ExPASy, GDB, MEDLINE, etc. for specific data items when these requested hypertext links exist to these foreign databases. Figures 5 and 6 shows an example of some of these search results. Figure 8 shows the organization of some of the PDD's CGI programs and the local database and disk resources they require.

As discussed before in Data Capture, Readers fill out paper forms as they read the papers and extract the relevant information. These paper forms are next proofread, reviewed by the PDD Editorial Board and then entered using the `SimpleForm` CGI program that processes the data by mapping it to the RDBMS *schema*. There are two steps in entering data: *basic data* (references, proteins, protein assay) and protein disease *correlations*. Correlations may only be entered after the basic data have been entered, since data is selected from scrollable lists of basic data.

The data are entered in the simple Web browser fill-in

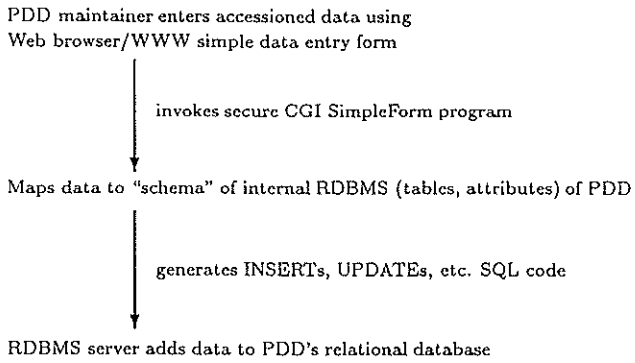

Figure 9. PDD data entry by mapping Web browser fill-in form data to internal RDBMS table data. Data from the paper form filled out by the Reader are checked and are then entered into the PDD by a simple data entry Web browser form. The form layout is user friendly. These form data are then received by the PDD which transforms it to the internal representation required for the relational database schema. At that point the data are inserted into the database.

forms either by typing in a text input window, cutting it from a scrollable review window and pasting it into the text input window, or selecting it from pull-down or scrollable menus (not shown). The SimpleForm program then translates the fill-in form data into the internal format required by the SQL schema used by the PDD to update the RDBMS tables and checkpoint them to disk files. Figure 9 shows the data mapping used in the PDD for entering data.

User specific "Selected Object Lists": One of the reasons we encourage users to register with the PDD is so they can keep small private lists of selected objects for a reasonable length of time for use in subsequent searches. Each user may create and keep *Selected Object Lists* between PDD sessions—one for each object type (protein, disease, reference). These lists of user data are kept on the PDD for a reasonable time after they are created or modified.

Selected objects lists may be used for several purposes. Search results of these objects are appended to the relevant list (e.g., proteins found are added to the Selected Protein List). In addition, proteins selected by clicking on spots in a 2-DE gel map are also appended to the protein list. When generating queries, a selected object list may be used to specify what to use in a search (cf. Figure 3). Once a list exists, users may then review, edit or specify objects that are in a list. For the list of proteins, this may generate a 2-DE gel map in the specified sample domain (e.g., plasma, CSF, urine) showing proteins currently selected. The list of independent variables mentioned above in querying the PDD database may be set from the selected objects list. So the list is used for both saving results and requesting searches.

C. Security issues for PDD, Web browser and the WWW

The current HTTP protocol used by the Web browser and WWW does not offer adequate security support without using the special versions of Web browsers that are not currently widespread. Security is important for several reasons, the most important being to protect the integrity of the PDD database, as well as, the computer systems on which they run.

We have designed the system to support several classes of users that are divided into three groups: system maintainers, PDD data maintainers and general users. The ability to change SQL statements is restricted to system maintainers. The ability to enter data is reserved for system and PDD data maintainers to protect database integrity. Furthermore, our goals of distributed maintenance of the database using Web browsers conflicted with using less secure versions of Web browsers more readily available (although this will change with time as more secure versions of these programs are offered). Therefore, it was necessary to implement a minimum level of security as part of our CGI programs for the PDD. When the PDD is first accessed, it requires a Username, Password and remote Internet "IP" host address (a default demo, demo is used if nothing is specified). It then repeatedly passes this and other user-specific parameters as time-encrypted information to later PDD operations.[7] These user specific parameters are passed from one PDD template expansion to the next, using HTML "hidden" encrypted variables. System and PDD data maintenance is further limited to PDD database maintainers by using protected templates limiting access to specific hosts and accounts. Protected directories are also employed where appropriate, using the standard HTTP server password authentication mechanism. As the WWW protocols security improves, we will be incorporating better methods to protect the database.

D. Dynamic HTML document synthesis using templates

To dynamically generate fill-in forms and process these form options, a dynamic HTML template language was developed. A template is a file with a .template file extension in HTML version 1.0 with a few new syntactic elements. It is expanded by the CGI program BuildForm into HTML using the latest information from the PDD's local databases. A query form in a HTML template may contain a synthesized conditional SQL query statement with various form-specific variables. This conditional SQL will later be evaluated by CGI program EvalSQLquery to generate the actual SQL query used in SQL query evaluation. Template operations include expansion of $form-variable$, conditional expansion of SQL Boolean subexpressions, multi-list SQL Boolean variables, as well as, including other template or HTML files.

The $form-variable$ instances in the SQL statement are expanded to their corresponding values specified by the user in the query form. When the SQL statement is evaluated by the PDD, the form conditional expressions {+ ...}+ or {− ...}− are used to modify (include or omit) that part of the SQL query where the conditional form-variables exist or not respectively.

[7] So, attempting to access intermediate PDD forms in your Web browser "hotlist" or "bookmarks" at a later time will not work.

Table 5 Example of evaluation of SQL query in template. This illustrates evaluation using the $. . .$ form-variable syntax. The initial SQL query statement is given in a). Let the HTML check-box settings returned to the PDD server be use_source_name="off" and use_fold_change="on", and with fold change range values between 2.0 and 5.0. Then, EvalSQLquery would expand the code to the SQL shown in b) that is now of the form required by the SQL server

a) Initial SQL query statement

```
SELECT protein.prot_id, protein.prot_name,
       dis_correl.fold_change, condition.cond_name
  FROM protein, dis_correl, condition, source
  WHERE
{+ use_source_name
   (dis_correl.source_id = source.source_id) AND
   (source.source_name = '$source_name$') AND
}+
   ((
{+
   use_fold_change
    (dis_correl.fold_change .GT. $fold_change_lower_bnd$) AND
    (dis_correl.fold_change .LT. $fold_change_upper_bnd$) AND
}+
   (condition.cond_name = '$cond_name$') AND
   (condition.cond_id = dis_correl.cond_id) AND
   (dis_correl.prot_id = protein.prot_id))
   )
```

b) Expanded SQL query statement

```
SELECT protein.prot_id, protein.prot_name,
       dis_correl.fold_change, condition.cond_name
  FROM protein, dis_correl, condition, source
  WHERE
    ((
     (dis_correl.fold_change > 2.0) AND
     (dis_correl.fold_change < 5.0) AND
     (condition.cond_name = 'Bladder Carcinoma NO') AND
     (condition.cond_id = dis_correl.cond_id) AND
     (dis_correl.prot_id = protein.prot_id))
    )
```

There are also cases where the Web browser form submitted to the PDD WWW server may supply multiple instances of a form variable. These multi-list variables are used in {&. . .}& or {|. . .}| expressions are expanded to multiple relational clauses in the "WHERE" Boolean expression of the SQL query statement. Finally, the user may optionally view the resulting SQL query statement template. Only privileged users are allowed to modify SQL query statements in the forms to protect the integrity of the PDD database. This is enforced by computing a time-stamped sequential checksum function of the SQL query statement by the BuildForm program and later comparing this checksum with the EvalSQLquery program when the SQL is returned for evaluation. We now illustrate the syntax of the template language by examples and with a complete example of a SQL query in Table 5.

PDD query template language syntax

Form-variables: During processing of the form SQL statement, SQL variables are evaluated from form-variables as part of the Boolean search condition for the SQL Select statement. If the form-variable NAME="$cond_name$" is set to 'Bladder Carcinoma' in the form, then

```
(condition.cond_name = '$cond_name$') AND
```

is mapped to

```
(condition.cond_name = 'Bladder Carcinoma')
AND
```

Conditional expressions: Conditional expression is used in the SQL query in the template to add or delete SQL expressions from the Boolean condition. The "include" condition uses the SQL code if the form-variable exists

```
{+ form-variable
   . . . SQL code . . .
}+
```

and, the "omit" condition uses the SQL code if the form-variable does not exist

```
{- form-variable
   . . . SQL code . . .
}-
```

Multilist form elements: HTML is used by the Web browser/WWW to provide multiple instances of a form variable with different values. Such Multilist form elements may be expanded in the SQL query template as

```
{&
   (user_value .EQ. '$user_values$') AND
}&
```

For three variables, this would expand to a "conjunction" of clauses:

Table 6 Support for HTML form menus in PDD templates. Part a) shows the types of template syntax and the resulting HTML that is generated. For menus, the NAME="value" used for the generated ⟨SELECT⟩ statement is set to NAME="table.attribute". The table.attribute data from the RDBMS are used to generate ⟨OPTION⟩ entries. Part b) shows additional menu request modifiers that may be added for additional functionality

a) Template syntax and the resulting HTML that is generated
   1. *table.attribute*
      generates a single selection HTML ⟨SELECT⟩ pull-down menu,
   2. @table.attribute@
      generates a single selection scrollable ⟨SELECT⟩ window,
   3. @table.attribute@+
      generates a MULTILIST selection scrollable ⟨SELECT⟩ window,
   4. #table.attribute#
      generates a scrollable non-selectable HTML ⟨TEXTAREA⟩ window for reviewing data.

b) Menu generation modifiers may be used for additional functionality. Let ? be one of the above menu type indicators (i.e., *, @ or #).
   1. ! appended in the ?table.attribute?! syntax sorts the ⟨OPTION⟩ entries by the most recent RDBMS edit_date for that table's entries. The alternate form !"sortTable.sortAttribute" will sort it by a different entry.
   2. ="*new-name*" used in the ?table.attribute?="new-name" syntax specificies the NAME field with the NAME="new-name" instead of the default NAME="table.attribute" in the ⟨SELECT⟩ statement. This syntax allows using several copies of the same menu for different purposes in the same form.

```
((user_value = "var1") AND
 (user_value = "var2") AND
 (user_value = "var3")
 ) AND
```

and,

```
{|
   (user_value .EQ. '$user_value$' AND
|}
```

would expand to a "disjunction" of clauses:

```
((user_value = "var1" OR
 (user_value = "var2") OR
  user_value = "var3")
 ) AND
```

Support for menu generation in HTML forms for PDD templates

Users should be able to interact with the latest RDBMS data in the HTML forms presented to the Web browser user. Therefore, we developed a menu generation syntax as part of the template language. PDD template-expansion using the BuildForm CGI program supports dynamic forms menu generation using current RDBMS data. The menu generation language uses RDBMS data matching any "table.attribute" in the database. The following template language syntax shown in Table 6 can generate a variety of menus presented as pull-down menus, single and multiple instance selection lists, etc.

Other HTML template operations

It is also useful to include other template or HTML files at arbitrary points in a template. The #include:*file*# syntax inserts the specified file's contents at that point in the template file currently being expanded. This allows subsets of documents to be reused consistently in several places. And finally, $form-variables$ may be used anywhere in the text (in addition to the SQL query in the templates) as a general macro-expansion variable. These make it easy to use one template to generate several different HTML forms based on selecting a particular option. For example, selecting a sample source could be used in selecting 2-DE gel map images involving that sample source in the generated form.