Report of the Workshop on Interconnection of Molecular Biology Databases

Held in Stanford, California, August 9–12, 1994

Edited by:
Peter D. Karp
Artificial Intelligence Center
SRI International
333 Ravenswood Ave., EJ229
Menlo Park, CA 94025
pkarp@ai.sri.com

This workshop was sponsored by grant IRI–9223350 from the National Science Foundation, and by the Biomatrix Society.

SRI International Artificial Intelligence Center Technical Report SRI-AIC-549.

Abstract

There is tremendous synergy between the roughly 100 existing molecular-biology databases. Once these databases are interconnected, biologists will be able to integrate diverse sources of information to answer questions that are laborious or impossible to tackle today. This NSF-funded workshop brought bioinformatics researchers and users of molecular-biology databases together with computer-science specialists in database interoperation. The workshop surveyed existing molecular-biology databases and the requirements for interoperation among them. Computer scientists presented an overview of the database-operation problem, and of techniques for solving it. Participants described a wide range of approaches to interoperation of molecular-biology databases, that are generating practical results. Existing systems allow multidatabase queries to databases such as Genbank, GDB, and PDB. There now exists no single, final resolution to the interoperation problem. Current approaches differ along a variety of dimensions including ability to handle complex queries, difficulty of implementation, required user expertise, and scalability. An understanding of these dimensions is important when deciding what techniques to employ for a given scientific community. The workshop identified a number of barriers to interoperation, such as resistance to standards, inaccessibility of existing databases to structured query via Internet, and poor documentation of many databases. But interoperation is proceeding at a rapid pace that promises to fuel scientific discovery.

Chapter 1

Introduction

There is tremendous synergy between the roughly 100 existing molecular-biology databases (DBs). Once these databases are interconnected, biologists will be able to integrate diverse sources of information to answer questions that are laborious or impossible to tackle today. Indeed, one could argue that bioinformatics faces an integration crisis because much of the value of these data is squandered by their current isolation.

Interoperation of heterogeneous databases is a hot research topic in computer science. Newly developed computer-science techniques can be applied to the problem of interoperation of molecular-biology databases. And because molecular-biology databases have complex semantics, and utilize diverse data models and platforms, they provide a rich testbed for computer-science research.

A workshop entitled "Interconnection of Molecular Biology Databases" was held at Stanford University on August 9–12, 1994. The workshop was sponsored by the National Science Foundation and by the Biomatrix Society, and was organized by Dr. Peter D. Karp of the SRI International Artificial Intelligence Center, with program committee assistance (Appendix A). The meeting brought together 55 bioinformatics researchers, computer scientists, and biologists from nine countries (see Appendix B for a list of participants). The participants included members of genome centers at Baylor College of Medicine, the University of Pennsylvania, the Whitehead Institute, Lawrence Berkeley Laboratory, Lawrence Livermore National Laboratory, Genethon, and the Sanger Centre.

The four meeting days included both formal presentations to the entire workshop, and informal discussions in smaller working groups (Appendix C gives the meeting agenda). This report provides an overview of the workshop, and summarizes both the formal presentations and the working group sessions.[1]

1.1 Dissemination of Workshop Results

To maximize the dissemination of the results of this workshop to the scientific community, we have prepared a set of World Wide Web (WWW) documents describing the workshop, rather than the traditional photocopied booklet of abstracts.

[1]This report was authored jointly by Peter Karp, the program committee, and a representative of each working group.

Scientists can use the Internet to access the following hypertext documents from the SRI International Web servers starting at URL `http://www.ai.sri.com/people/pkarp/mimbd.html`:

- The call for participation and summary of workshop goals
- Abstracts from the workshop participants
- The meeting agenda
- Contact information for workshop participants
- A summary of biological databases, including Web pointers to many of them
- This summary report of the workshop

These Web documents will be publicized to the scientific community through a number of channels, including those used to publicize the workshop.

1.2 Overview

The range of issues discussed at the workshop included specifications of the database interoperation problem, potential solutions to that problem, the benefits of interoperation, and barriers to interoperation.

1.2.1 Importance of Database Interoperation

What benefits will result from solution of the interoperation problem? For a variety of reasons, the value of an integrated collection of molecular biology databases is greater than the sum of the values of each member of the collection. Therefore, without integration, the value of these databases is diminished. Databases are more useful when integrated because

- Biological data are more meaningful in context and no single database supplies all context for any datum. For example, we better understand a gene when we know the function of its product, the sequence of the gene and its regulatory regions, the three-dimensional structure of its products, and the functions of evolutionarily related genes. These types of information are scattered across different databases.
- New biological theories and regularities are derived by generalizing across a multitude of examples, which again are scattered across different databases.
- Integration of related data enables data validation and consistency checking.

1.2.2 The Database Interoperation Problem

The longer one works at defining the problem of interoperation of molecular-biology databases, the broader the definition becomes. Ideally, an integrated collection of databases should allow a user to interact with every member of the collection, and with the collection as a whole, as seamlessly as they can now interact with any single database. The word "interact" includes many modes of use: general browsing, seeking information about particular objects, performing complex queries and analytical computations, submitting new experimental results to community repositories, and curating a collection of information.

The users of molecular-biology databases have widely varying expertise with respect to both computers in general, and to knowledge of particular databases. Users should be able, but should not be forced, to bring expert knowledge to bear on an interoperation problem. For example, a user who knows which databases to query should be allowed to specify them; otherwise the system should determine relevant databases. The system should also tag each data item with its source if the user so desires.

Biological databases span a wide range of database technologies, so in the general case we require interoperation across relational, object-oriented, flat-file, and various "home-brewed" database systems. However, attendees of this workshop showed a definite movement toward object-oriented systems. This trend may or may not be typical of the bioinformatics field as a whole, but if it is typical, it simplifies the interoperation problem.

1.2.3 Solutions to the Interoperation Problem

Currently there exists no single, final resolution to the interoperation problem. The participants of the workshop put forward a number of different solutions, each of which has different strengths and weaknesses. Quite simply, one cannot begin to grasp the field of database interoperation without understanding the trade offs that exist among the available solutions. If different communities of biologists have different requirements for interoperation, different approaches may well be optimal for each community.

The dimensions along which interoperation techniques differ are

- Performance
- Difficulty of implementation
- Ability to handle complex queries
- Ability to handle textual versus structured data
- Resilience to schema changes in member databases
- Degree of user knowledge required about:
  - Schemas of all member databases
  - A special global schema
  - What databases contain what types of information
  - Physical locations and access mechanisms for member databases

- Ability of users to update member databases
- Ease and timeliness with which updates to member databases are available to users
- Reliance on use of standards within the bioinformatics community
- Scalability

For example, the approach of interoperation through physical integration, in which member databases are translated into a common schema and physically loaded into a single database management system (DBMS), is fairly easy to implement (although it requires many data translators), performs well, can handle complex queries, requires little user knowledge of member database schemas or locations or access mechanisms, and does not depend on adoption of standards. But this approach does not scale well, is not resilient to schema changes, requires user knowledge of the global schema, does not allow user updates, and can be slow to incorporate updates from member databases.

In contrast, the mediator approach scales well, is resilient to schema changes, allows immediate access to member-database updates, allows user updates, can handle complex queries, minimizes user knowledge of database locations and schemas and access mechanisms, and does not rely on adoption of standards. But performance is subject to network delays, and the strategy is complex to implement and is still a research topic.

The contrast between these approaches illustrated a significant conflict at the workshop: the desire to employ old, well-understood techniques that can yield limited solutions very quickly, versus a reliance on newer techniques that are more complicated and still require research, but could yield more flexibility and power.

The danger of the first approach is that existing techniques have known and probably unknown limitations, which if unheeded and unanticipated could lead to an unworkable system that collapses under its own weight as its size and requirements increase. The danger of the second approach is that unproven ideas may never pan out. Funding agencies must seek a proper balance between these investment strategies.

As one would expect, the earliest results of interoperation have been achieved using the first type of approach (see, for example, summaries of talks by Ritter and Etzold). It is worth noting that more complicated techniques that are still undergoing research are also yielding practical results (see summaries of talks by Overton and Arens).

A surprisingly large number of groups are working on the interoperation problem. However, the problem is so big that we cannot expect any one group to develop a solution. It may be more appropriate for different groups to contribute components of an overall software architecture for database integration.

1.2.4 Barriers to Interoperation

A number of nontechnical barriers to interoperation were identified at the workshop:

- Although standards at a variety of levels can facilitate interoperation tremendously, workshop participants expressed strong resistance to standards, in part out of concern that standards

stifle creativity, and because significant efforts are often required to modify existing software to conform to standards.

Yet the fundamental value of standards would seem to be indisputable — witness the success of the Internet and the WWW, which would be impossible without standardization. One way to account for this paradox is to note that different people employ different definitions of the term standard: (1) a well-documented convention prescribed by a higher authority to control all aspects of an interaction between organizations, or (2) one of a family of well-documented conventions adopted by mutual agreement of a group of organizations to control some well understood subset of their interactions. The latter definition, based on voluntary adoption of rules governing interactions that are so well understood as to require no significant creativity, is probably more appropriate for research organizations. It is worth noting that interoperation without standardization is impossible according to the latter definition: whether we wish to call them standards or not, interoperation is impossible without shared conventions.

- Few incentives now favor interoperation; funding and scientific credit often reward efforts that distinguish themselves according to how they differ from prior work, rather than according to their compatibility with prior work.
- Many existing molecular biology databases are not accessible via Internet query; similarly, many biologist users do not have Internet access.
- The semantic descriptions of many molecular-biology databases are terribly incomplete. Without an understanding of the semantic relationships among databases, interoperation is impossible.

1.2.5 Caveats of Interdisciplinary Research

As an interdisciplinary field that spans computer science and molecular biology, the area of interoperation of molecular biology includes the usual drawbacks and advantages of interdisciplinary research.

Computer scientists benefit from a challenging real-world domain in which to test and refine their techniques, but they should beware of oversimplifying the requirements of these complex problems, of supplying solutions to problems that do not exist, and of providing elegant solutions that do not work in practice.

Biologists can seek answers into previously unattainable questions to obtain deeper insights to biological systems, but should beware of overlooking known limitations of existing techniques, and of overlooking newly developed techniques.

Each group should respect the research goals of the other, seek to bridge differences in language and research culture, seek to understand what constitutes a research result in this interdisciplinary field, and know where results can be published.

Chapter 2

Summary of Workshop Sessions

2.1 Tuesday Sessions

2.1.1 Introductory Session

The first day of the workshop began with a series of introductory talks on molecular-biology databases, and on database interoperation.

R. Smith: "Overview of Molecular-Biology Databases"

Smith presented an overview of molecular-biology databases that surveyed the types of molecular-biology information typically residing in electronic sources (such as nucleic-acid sequences, amino-acid sequences, protein structures, genomic maps, and bibliographic data). He also listed specific databases that contain each type of data, such as Genbank, SwissProt, PDB, GDB, and Medline. Smith's catalog of databases is accessible online from the MIMBD WWW server.

S. Davidson and X. Qian: "Survey of Computer-Science Approaches to Database Interoperation"

The database interoperation problem involves providing integrated access to a collection of preexisting databases. Interoperation is a difficult problem because different databases employ different data models and different query languages, and because different databases may have schema-level conflicts and conflicts of fact. Computer-science researchers have addressed many aspects of the interoperation problem, including

- Architectures for database integration
- Techniques for merging schemas and detecting schema conflicts
- Translation among query languages
- Languages for multidatabase queries
- Optimization of multidatabase queries
- Updates to integrated databases

Architectures for database integration range from a loose to a tight coupling. A tightly coupled architecture makes use of a global schema that conceptually integrates the schemas of all member databases. The advantages of this approach are that it simulates a homogeneous distributed database and provides explicit resolution of schema-level conflicts and conflicts of fact. The disadvantages are that effort is wasted if all conflicts are resolved in cases when complete integration may not be necessary, that schema integration must be repeated each time the schema of a member database changes, and that the system cannot tolerate the loss of a member database.

A loosely coupled architecture expands the data definition language to allow member databases to import/export data from/to other member databases. The query language is expanded to allow queries to extend to any imported schema. This approach avoids the expense of complete integration and allows user queries to range over multiple databases, but it provides no help for resolving conflicts of schema or of fact, and provides no assurance that query results are meaningful because of these unresolved potential conflicts.

Davidson described her group's work on approaches to schema translation and automated schema merging (the former is expanded upon by Kosky's presentation).

General surveys of computer science research on database interoperation can be found in: "Integration of Information Systems: Bridging Heterogeneous Databases," edited by A. Gupta, IEEE Press (1987); ACM Computing Surveys 22:3 (Sep 1990); IEEE Computer 24:12 (Dec 1991).

Qian's portion of the survey began with a discussion of the dimensions of semantic heterogeneity (the schema conflicts discussed by Davidson). She enumerated ways in which meanings of schema definitions for similar types of concepts might differ and then discussed the implications of such differences, and strategies for resolving them. Two databases for genetic-mapping data might have three types of meaning heterogeneity:

- Granularity — In one database the gene-to-sequence relation might be one-to-one, whereas in the other database it is one-to-many
- Scope — One database might explicitly encode the scope of its data with respect to species, whereas in the other database species is implicit because the database covers only one species
- Temporal basis — The databases might employ different notions of time

The databases might also contain representation heterogeneity of several types:

- Different names for equivalent tables
- Different table structures for equivalent information
- Different choices as to whether the same concept is encoded as an entity or a relationship
- Different primary keys for the same tables

Knowledge of these semantic relationships is important when querying multiple databases, when interpreting results of multidatabase queries, and when performing schema integration. If, for example, a user wishes to apply a query to five different genomic-map databases that employ five different relational table structures for encoding map data, either the user or the multidatabase query system

must have enough understanding of these different table structures to formulate appropriate queries to each different database. Qian surveyed the extent to which several computer-science strategies for multidatabase querying shield the user from having to understand and grapple with semantic heterogeneity when querying multiple databases and interpreting the results of those queries.

The techniques for multidatabase querying are

- Allow queries to be formulated in a multidatabase language in which all table names are qualified with database names – that is, a query explicitly references tables in multiple databases.
- Queries are formulated in a federated schema; the database system uses a view mechanism to automatically translate the query between the federated schema and the schemas of each member database.
- Queries are formulated the schema of any member database, and are automatically translated into the schemas of other member databases

The first approach puts the full burden of understanding and reconciling semantic differences on the user. The second approach takes that burden off the user, but requires full schema integration and user knowledge of the federated schema. The third approach is extremely promising because it relies on only partial knowledge of relationships among schemas, and allows users to formulate queries in terms of schemas with which they are already familiar. Qian is pursuing the third approach in her work on mediator architectures.

2.1.2 Session on Requirements for Database Interoperation

R. Robbins: "Community Databases: Towards a Federated Information Infrastructure"

Robbins's thesis is that whereas data acquisition was the genome-informatics crisis of the 1980s, data integration is the genome-informatics crisis of the 1990s. Scientists are currently unable to answer straightforward queries that span multiple genome databases, thus failing to realize much of the value of these data. The informatics community must achieve a conceptual integration of genome data in concert with analytical software. Robbins argued that a loosely coupled approach to database federation is a promising candidate architecture that is achievable with today's technology. The WWW is an example of this approach. Much of the success of the Web can be traced to its generic client-server architecture, in which a single client program offers users access to a multitude of servers. Robbins proposed an analogous Federated Object-Server Model (FOSM) for bioinformatics, where participating databases "publish" their data as read-only objects, represented in a standard data model. Generic client software would retrieve data from a read-only database federation. Requirements for the FOSM were outlined, and a tree-based algebra of database operations was proposed.

T. Slezak: "Database Interoperation"

Slezak outlined three requirements for database interoperation that genome laboratories must satisfy:

- Submitting summary data to community resources such as GDB
- Providing raw data access to genome-center collaborators
- Providing query access whereby users at the genome center can access community resources

Slezak advocated an extremely pragmatic approach to interoperation, arguing that existing database application program interfaces (APIs) allow him to solve all of his center's interoperation problems. He argued that low-level API-based approaches can handle all of the needs of the bioinformatics community; therefore, there is no need to consider or wait for higher-level, more sophisticated computer-science techniques, some of which are still in the research stage.

G. Wiederhold: "Intelligent Integration of Information"

Wiederhold introduced the notion of a mediator as an intelligent interface between a user (or an application) and a collection of databases. A mediator is a "fat interface" in the sense that it performs significant amounts of processing, such as resolving mismatches, finding relevant data, and reducing the quantity of final data presented to the user using abstraction and summarization. Wiederhold argued that without intelligent mediation, the growing trickle of molecular-biology data available over the Internet may soon become an overwhelming firehose.

A mediator is intermediate between the loose and tight forms of coupling discussed by Davidson. Rather than forcing full schema integration, it allows partial integration based on knowledge of database interrelationships. By allowing more autonomy of component databases than does tight coupling, the mediation approach facilitates greater scaling. For example, imagine a multidatabase system based on tight coupling that integrates the schemas of 25 databases. If each database undergoes only one schema change per year, the federated schema must be reintegrated every two weeks on average. A federation with more databases whose schemas change more quickly (likely in the molecular-biology arena because of its complex data) may require such frequent reintegration as to disintegrate.

M. Graves: "Conceptual Models: Not Just for Design Anymore

Conceptual models have traditionally been used for database design, but they can also be used to define a common data exchange language between databases. Many graph-based conceptual models have their genesis in natural language processing research. These formalisms which were originally designed to model human discourse and which have proven themselves capable for describing database schemas are also useful for exchanging data between databases.

The data exchange process is to create a common conceptual schema using a graph-based conceptual data model, develop a view of the schema for each database, decompose the schema into binary (dyadic) relations and develop a translator between each relation or object in the database and a set of binary relations. Creating a common conceptual schema may be a difficult process in general, but it is easier in the genome domain because genome data has a graph-like structure and a common conceptual schema has already been developed for several major molecular biology databases. It is fairly straight forward to develop views of the common schema for each database and to decompose the schema into binary relations. Developing a translator between the relations or objects being used and binary relations is not difficult but can be time consuming, so Graves has developed a graph logic programming language which allows the translation programs to be written as simple logic programs.

2.2 Wednesday Sessions

2.2.1 Database Interoperation Session

W. Klas: "Demand-driven Database Integration for Biomolecular Applications"

The presentation reported on a project that was driven by the needs of the Reliwe project and aimed at producing a working application in support of rational drug design, involving the management of receptor-ligand data and other relevant information. The database work required that (1) data from heterogeneous databases be integrated, (2) the integrated data be represented and interconnected within a structured data model, and (3) the system provide efficient answers to declarative queries against the final information resource. The project was faced with the usual problems, such as incomplete and inaccurate data. The goal of the system was to provide support for complex data in an extensible data model, to provide a flexible query language, and to integrate data that were heterogeneous in type, source, format, and content. VODAK was the database used. VODAK is a research prototype open OODBMS. The Volcano extensible query optimizer (developed at Boulder, Colorado) was used. The general solution framework was first to derive from each participating database an enriched export schema (EES), then to integrate the EESs into a single, global, integrated schema. The integration process was reported to be expensive and to include many one-way operations.

Although this project does involve data relevant to molecular biology, it also is a specific application that can best be thought of as providing an example of how such data might be integrated.

Y. Arens: "SIMS: Single Interface to Multiple Systems"

The SIMS project report described an ongoing effort at USC/ISI to facilitate retrieving and integrating data from multiple information sources using a "mediator" technology that involves developing an intermediate knowledge base (KB) that describes the multiple relevant databases, and that can facilitate integrated queries across them.

The SIMS approach to integrating heterogeneous information sources involves developing a domain model of a problem domain, in this case transportation planning. The domain model is encoded in a knowledge representation language (Loom).

SIMS accepts queries in the form of a description of a class of objects about which information is desired. The description is composed of statements in the Loom knowledge representation language. The user is not presumed to know how the data are distributed across the underlying databases, but the user is expected to be familiar with the application domain and to use standard domain language to compose the Loom query. An overview of SIMS procedures are shown in Figure 2.1.

Additional information can be found in Arens, Y., Chee, C. Y., Hsu, C-N, and Knoblock, C. A. 1993. "Retrieving and integrating data from multiple information sources". *International Journal of Intelligent and Cooperative Information Systems*, 2:127-158.

Note that this project does not presently involve molecular-biological data and thus is best seen as another example of potentially relevant methods.

O. Ritter: "The IGD Approach to Interconnection of Genomic Databases"

Ritter reported on his project to integrate data resources relevant to the human genome project. He developed mapping rules for public domain databases such as GDB, PDB, and SWISS-PROT,

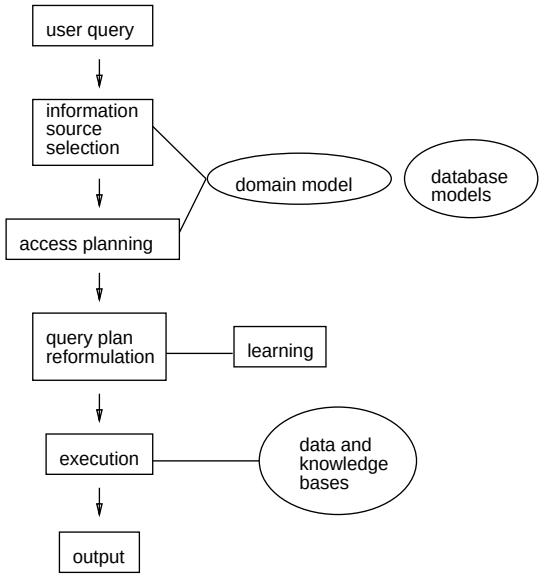

Figure 2.1: The SIMS strategy for processing multidatabase queries.

through which data in these databases are periodically collected, reformatted, and imported to IGD. The technique has produced usable browser tools into genome data at a level of effort that has proven to be much less than many observers had expected.

This approach shows that, in the right hands (provided those hands are willing to use many pre-existing tools) it is possible to carry out highly useful integration of data resources into a single, coherent browsable whole. The problem of integration and interoperability to the point of joint updates across multiple resources is still problematic. It is also not clear that this method, however amazingly cost-effective it has been at the DKFZ, would scale to any general extent.

According to Dr. Ritter's experience, the cost of incorporating new data sources is reasonable.

C. Overton: "Using a Query Language to Integrate Biological Data"

The group from the University of Pennsylvania reported on an approach to information resource integration involving the development of tools to support a high-level query language capable of merging both data and computing requests in single queries. This is work at the level of infra-infrastructure and as such must be considered as a potential component in a larger context.

The goals are to develop a (read-only, at least initially) query language to support ad hoc queries against a variety of heterogeneous information resources. The problems faced by the project have included (1) a wide variety of underlying data models and systems, (2) retrieval interfaces into existing systems that are often complex and entangled in application programs, (3) the need to integrate analyses with data requests, and (4) schema differences of varying subtlety. The requirements have included the need to (1) provide an expressive type system, (2) allow for query translation, (3) offer extensibility, and (4) have some query optimization.

The underlying type system brings relational, ASN.1, ACE, and OODBMS data objects into a common notion of "collection." The Collection Programming Language (CPL) then can be embedded in user application programs to allow collective queries to be run against the underlying integrated information space spanned by participating host systems. A model view of the system is shown in

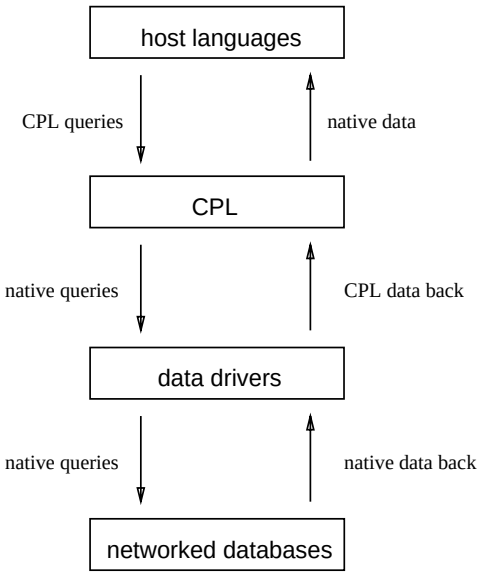

Figure 2.2: Query processing using the CPL (queries go downward on the left and replies return upward on the right).

Figure 2.2.

R. Durbin: "Dynamic Links between AceDB and Molecular-Biology Databases"

The AceDB is used to manage genome sequence and map data for the *C. elegans* genome project. Durbin described mechanisms for creating links from AceDB databases into other databases, such as SWISS PROT. The method relies on a convention for naming objects in remote databases using a syntax of the form `database:id`, where `database` is the name of the database, and `id` is the unique identifier of an object in that database. These remote-object names are stored in regular fields of AceDB objects, where software can access them and retrieve the specified object from the specified database.

N. Goodman: "The Case for Componentry in Genome Information Systems"

Goodman advocates the construction of genome information systems using modular, "plug-and-play" components. Examples of components include significant subsystems such as databases, analysis programs, and user interfaces. Current practices at most genome informatics centers are to either build systems from scratch (which is expensive, and stifles creative progress in the field because of constant duplication of effort), or to adopt a complete existing system (such as AceDB) and use it with minor customizations (forcing the informatics center to live with all of its quirks and limitations). There is no middle ground in which a designer can build some parts of the system and adopt existing components for the remainder.

To make the "unit of contribution" in genome informatics become the component, rather than the complete system, we require an architectural framework that facilitates software reuse. This framework must include the following features:

- Low-level representatives and protocols
- Definitions of molecular-biology object types that are extensible

- Ensuring object interoperability between Unix, Windows, and Macintosh computers
- Allowing object transfer by use of e-mail, ftp, Gopher, WAIS, and WWW
- Defining protocols that let users invoke behaviors associated with molecular-biology objects

If any framework is to be widely accepted by the bioinformatics community it must be the result of a community-based effort that includes substantial consensus.

P. Karp: "Encyclopedia of E. coli Genes and Metabolism"

The goal of the EcoCyc project is to compile a large knowledge base (KB) describing the genes and intermediary metabolism of E. coli. The KB will describe each pathway and bioreaction of E. coli metabolism, and the enzyme that carries out each bioreaction, including its cofactors, activators, inhibitors, and the subunit structure of the enzyme. When known, the genes encoding the subunits of an enzyme will be listed, as well as the map position of a gene on the E. coli chromosome. In addition, the KB describes the chemical compounds involved in each bioreaction, listing synonyms for the compound name, the molecular weight of the compound, and in many cases its chemical structure.

The EcoCyc KB is designed as an electronic reference source for E. coli biologists, and for biologists who work with related microorganisms. Using the EcoCyc graphical user interface, scientists can visualize the layout of genes within the E. coli chromosome, or of an individual biochemical reaction, or of a complete biochemical pathway (with compound structures displayed). Navigation capabilities allow the user to move from a display of an enzyme to a display of a reaction that the enzyme catalyzes, or of the gene that encodes the enzyme. The interface also supports a variety of queries, such as generating a display of the map positions of all genes that code for enzymes within a given biochemical pathway. As well as being used as a reference source to look up individual facts, the EcoCyc KB will enable complex computations related to the metabolism, such as design of novel biochemical pathways for biotechnology, studies of the evolution of metabolic pathways, and simulation of metabolic pathways.

EcoCyc currently interoperates with Medline by dynamically retrieving publication and abstract information using the NCBI toolbox. The longer term goals of the project are for EcoCyc to interoperate with other biological databases containing protein and nucleic-acid sequence data, protein structures, descriptions of protein products other than enzymes, and genetic regulatory data. Karp plans to use EcoCyc as a testbed for investigating mediation architectures for interoperation.

2.3 Thursday Sessions

Session on Data-Definition Tools

V. Markowitz: The Object-Protocol Model and OPM Data Management Tools

Markowitz has developed an Object-Protocol Model (OPM) and OPM-based data management tools. OPM combines in a unified framework traditional object constructs with special protocol constructs for modeling scientific (e.g., molecular biology laboratory) experiments. Furthermore, OPM provides constructs for specifying views, inter-database references in a multidatabase environment, and versions.

OPM data management tools facilitate the development of genomic databases using commercial relational database management systems (DBMSs) and allow scientists to define, query, and browse genomic databases in terms of application-specific objects and protocols. The OPM tools currently target relational DBMSs and will be extended to object-oriented DBMSs. The suite of OPM data management tools currently include:

1. A reverse engineering tool for determining the OPM description of existing relational databases
2. A graphical editor for specifying OPM schemas
3. A translator that maps OPM schemas into DBMS database definitions and procedures
4. A graphical browsing and data entry tool for browsing and updating OPM databases

OPM and the OPM data management tools are currently used for developing several genomic database systems, such as version 6 of Genome Data Base (GDB) at Johns Hopkins School of Medicine, Baltimore. Markowitz plans to use OPM for constructing a federation of genomic databases, where OPM will be used as the common data language for describing the database members of the federation.

OPM documents, papers, and data management tools are available via World Wide Web using URL: `http://gizmo.lbl.gov/DM_TOOLS/OPM/opm.html`.

J. Gennari: Ontolingua

Ontolingua has been developed as part of the Knowledge Sharing Initiative within the artificial intelligence community for the purpose of developing portable *ontologies*. An ontology is a specification of a conceptualization — a written encoding of a representational vocabulary. A database schema is an example of an ontology. The hypothesis behind Ontolingua is that database interoperation will be easier if the databases to interoperate make use of shared, common ontologies (such as conceptualizations of genetic maps), than if each database group develops its own idiosyncratic ontology. Adoption of shared ontologies will result from a process of consensus building in which bioinformaticians examine the ontologies used for genetic maps (for example) in a variety of genome databases, and then construct one (or a few) ontologies for genetic maps that incorporate the range of representational subtleties discovered by the individual database designers. The designer of a new genetic-map database will then be able to pull an existing ontology "off the shelf," modify it as needed for that database, and convert it into the data-definition language used by the DBMS used for that database. The fewer modifications that are made, the more easily this database will interoperate with other genome databases based on the same ontology.

Ontolingua could have additional uses in database operation besides encoding shared ontologies. For example, the task of querying a set of genome databases that employ different ontologies is simplified if we have a common data model that spans the data models of those databases. Ontolingua has the power to serve as a common data model because it unifies the relational and object-oriented models: the relation and the class are both primitive constructs in Ontolingua.

More information on Ontolingua, and WWW encodings of several sample ontologies, can be obtained at URL `http://www-ksl.stanford.edu/knowledge-sharing/README.html`.

Database Session

In this session, representatives from a number of database projects reported on their databases focusing on those aspects of the work that involve interactions with other database centers and the general user community, and that touch on issues of large-scale database interoperability. These projects were presented as working examples of the feasibility and relative merit of particular strategies for achieving global database integration.

E. Barillot: "The IDB Database System and Its Use in the Human Genome Project: HUGEMAP"

HUGEMAP is an example of a database that integrates a wide variety of data with disparate characteristics. It includes Genethon's and CEPH's human physical mapping data, an integrated map of the human genome, part of Genethon's genetic mapping data and a cytogenetic description of the human genome. Also included are external (prepared outside of the Genethon/CEPH project) physical mapping data and the CEPH directory of collaborative world-wide research on the genetic map. Methodology for the correlation of cDNA production and screening results, of cytogenetic translocation data, and of sequence data (including GenBank) is under development. Two million objects are currently stored in the database.

An object-oriented database management system, The Integrated Database System (IDB), was developed to manage the data, accommodate the specialized characteristics of the data, and to provide a query interface that incorporates data analysis within query operations. The system has been implemented over a specialized (home-made) storage manager; it offers standard database features such as reliability, security, and integrity constraints as well as object-oriented features such as support for abstract types (including methods) and type polymorphism. A specialized query language, Tcl, that supports an API has been developed and implemented. The system includes a generic browser that facilitates navigation through the HUGEMAP data collections. The system is available via WWW and through an e-mail server. A new release of IDB, built on a client-server architecture and supporting version management, is currently under development.

P. Bucher: "The Eukaryotic Promoter Database (EPD) and Its Relationship to the EMBL Nucleotide Sequence Data Library"

The Eukaryotic Promoter Database (EPD) contains a set of POL II promoter sites. These sites are represented as regions mapped onto nucleic acid sequences as represented in the EMBL nucleic acid sequence database. As an operational definition, promoter sites are considered to be equivalent to transcription initiation sites. The database also includes information on promoter regulation and experimental methods employed in identifying the site. Citations to the primary literature are included. The primary source of information for EPD is the published literature. Information extracted from the published literature is reviewed, interpreted, and corrected by the database staff.

EPD employs a strategy of very tight integration with the EMBL database. EPD sites are linked directly to subsequences in the EMBL database. The sequence data are not stored directly in EPD.

Information in the EMBL database is frequently revised. If not effectively propagated, such changes compromise the integrity of the links and the EPD data itself. Consistency is maintained manually, requiring close collaboration and strict update scheduling. This solution works in this environment because the EPD data set is relatively small and an effective collaboration has been maintained. The approach will not scale and will breakdown completely if the working relationship between the EMBL and EPD is not continued. No solution to these potential problems was proposed.

GenBank and the EMBL maintain a policy of exchanging data and producing two overlapping but independently compiled nucleic acid sequence data sets. The transformations of the data made at GenBank destroy the integrity of the EPD links, and it has proven to be too labor intensive to maintain a correspondence between EPD and the transformed sequence data in GenBank.

O. White: "EGAD: Expressed Gene Anatomy Database"

The Institute for Genome Research (TIGR) has launched an effort to experimentally determine the sequences of Expressed Sequence Tags (EST) corresponding to the complete set of human mRNAs (additional projects involve determining EST sequences from other model organisms). ESTs are relatively small sequence segments that serve as mRNA markers; they correspond to small fragments (subsequences) of the tagged mRNA. Analysis of these data can aid in identifying the function of the product of the tagged mRNA. An important method of analysis is the comparison of the EST sequences with those available in the public sequence databases, such as GenBank and PIR.

EGAD was developed to associate identified ESTs with information found in other macromolecular databases. It can be considered to be a database of linkages. The primary effort involved in maintaining EGAD is to ensure the integrity of the linkages, specifically the scientific integrity.

The EGAD project immediately encountered a major difficulty that will be faced in many database integration projects: how to handle fundamental differences in the levels of quality assurance and data integrity exhibited by various databases. Databases with low requirements on semantic consistency can be linked smoothly to those with higher requirements; however, the reverse is not true.

The EGAD researchers have set the following goals for their database

- The database should function correctly
- Success depends on a high-level of semantic precision and commitment to high quality
- Biological correctness is essential

Serious difficulties were encountered in attempts to establish links between EGAD and GenBank. The problems stem from GenBank's low requirements on semantic consistency. GenBank's data are inconsistent, incomplete, and often incorrect. Inconsistencies among data within GenBank lead to inconsistency among results of logically equivalent queries. As a result, it was difficult and in some cases impossible to establish the integrity of potential links.

A more careful specification of the semantics of GenBank would aid in establishing such links. However, ontologies are of limited value without robust data.

J. Blake: "STS: Sequences, Taxas, and Sources"

STS is a related project at TIGR. This data set exhibits the following characteristics:

- Each entity accessioned
- Satisfies minimum requirements for interoperability
- Semantics defined
- Controlled vocabularies

- Representations of sequences
- Alignment
- Links to diverse databases

An attempt to link this data set to the public macromolecular databases encountered problems similar to those described above. This presentation focused on the need for the establishment of, and the adherence to, controlled vocabularies. Even in projects where controlled vocabularies are adhered to, the vocabularies change rapidly with the addition of new information. Maintaining self-consistency within evolving vocabulary schemes is a nontrivial problem. The problem has been compounded by the lack of effective coordination among the various groups attempting to maintain such vocabularies. This results directly in dispersion and serious semantic inconsistency between databases.

D.G. Shin: "Designing an Interface Capable of Accessing Heterogeneous Autonomous Databases"

Johns Hopkins University is in the process of dramatically changing the operating conditions and underlying model of the database Genome Data Base (GDB). A model of GDB as a collection of heterogeneous autonomous data sets was presented briefly. The presentation focused on the interaction of users with this collection.

Users represent a broad class with distinct needs and requirements. To effectively satisfy their needs, user groups must be characterized and different strategies followed for different classes of users. The following classes of users were identified:

- Development staff
- Genome centers
- General users

The interface presented to the users must be tailored to their needs however, interfaces are intrinsically dependent upon the ability of the underlying data to support them.

Current work at GDB involves the development of effective mechanisms for restructuring queries for naive users. This involves resolving the discrepancy that inherently exists between the way users think the query should be evaluated and how the database model actually behaves.

Database Interoperation Session

This session focused on examining issues of database integration from the perspective of the user community. The general approach is to view the public data sets as read-only and to develop and/or navigate among external linkages among them.

T. Etzold: "SRS: an Integrated Query System for Molecular Biology"

SRS is an information retrieval system designed to access information across a large number of public macromolecular databases. Its design was based on the following assumptions:

- Most biologists need a mechanism for simple, quick database query that allows queries to be refined by successive queries connected using Boolean logic.
- Most databases in molecular biology employ similar flat-file formats for data distribution. These formats define a single, primary entity that is organized as a set of field-label field-value pairs.

In most of these databases, entries exhibit an additional internal structure: the field-label field-value pairs are organized within subentities. An entry is more properly understood as a nested set of entities, which contain attributes. It is typical in retrieval programs designed for these data to ignore this structure and to treat an entry as a single *document*. Data are indexed by *flattening* the entry and ignoring the implicit relationships among entities and attributes within the entry. SRS employs this strategy.

SRS accesses databases in the native formats distributed by the database providers. Relevant fields are indexed by parsing these formats. SRS provides a general solution to parsing flat-file formatted data sets. An extended-BNF language for describing format *templates* was designed and a YACC-like interpreter was developed for parsing formats described in this manner.

SRS includes link tables between databases–for example, links between entries in GenBank and PIR. These links are generated based on information provided in the databases. Links are defined to be unidirectional (from an entry in one database to any entry in another) but backward tracing is supported. Links are pairwise; many-to-one, and so forth; relations are flattened into sets of pairwise links.

The collection of databases included within SRS can be organized as a network of databases connected by *paths* defined by the links. SRS employs a query language that allows for navigation between databases by following the paths. When links are constructed in this manner, logically equivalent queries that follow different paths lead to inconsistent results. To a large extent these inconsistencies can be attributed to inconsistencies, incompleteness, and incorrectness in the link data supplied by the public databases. However, the simplifications imposed by the SRS link strategy also contribute: links among databases are more properly understood as links among subentities; ignoring these relationships leads to inconsistency.

SRS resolves path inconsistencies by selecting the shortest path between specified databases unless explicitly directed otherwise. A formalism is defined within the query language that permits paths to be explicitly specified by the user.

SRS is available via Mosaic; an API was developed while integrating SRS with the Mosaic interface. Like all other similar approaches, a detailed knowledge of the structure and properties of each database is required to effectively query it using SRS.

S. Letovsky: "Database Integration Using World Wide Web"

This presentation described three ongoing efforts involving database integration within the environment of the WWW.

The Genera project has developed an object-oriented data model for biological information. Tools have been developed for automatic generation of relational database schema from data described in Genera. The system employs Sybase as the relational engine. Genera provides a Mosaic interface operating directly over the Sybase server.

X-Locus is a database designed to interconnect a number of genetic databases. It consists of sets of links among these databases.

The remainder of the presentation focused on improving the directory structure of WWW and the ability of users to navigate across the web. The following recommendations were made:

- Site indexes should be eliminated: users are interested in topics not geographic locations.
- Indexes should be more widely shared.
- A network of dynamic macroreviews with links to resources should be established.
- A standardization of efforts is desirable.

A. Kosky: "TSL a Transformation Language for Integration and Evolution of Biological Databases"

Data transformation occurs in a number of settings within the general framework of data integration and the maintenance and evolution of integrated views. These include

- Rapid schema evolution
- Integration of distinct heterogeneous databases
- Conversion of captured data to its stored form
- Creation of multiple user views

TSL is a formal computer language that models data transformations as a collection of transformation operations defined as declarative statements. This approach fosters internal consistency, reproducibility, and formal completeness by formulating transformation processes as logical assertions. TSL employs a declarative, Horn-clause logic with nested relations. It unifies the concepts of transformations and database constraints in a single framework. Identities are modeled by skolem functions.

The following complications were encountered in this work:

- Incompatible data models
- Hierarchical database structures
- Nonstandard data integrity constraints

2.4 Friday Sessions

M. Zorn: "Meta-Driven User Interfaces"

Zorn addresses the problem of automatically generating graphical user interfaces (specifically, for now, X-Motif) to databases using metadata information. The "standard" GUI generated from this schema information can then be specialized for an application. The GUI can be used for querying as well as for updating. When used for data entry, users often wish to augment the metadata with

conventions (e.g., undeclared, implicit restrictions) and follow-up (triggered events for related fields). GUIs generated by Zorn's approach have been tested internally at LBL as well as for GSDB.

D. George: "Conceptual Design of Macromolecular Sequence Databases"

PIR is an international, multidatabase package (USA, Germany, Japan) whose partners cooperate on database design and documentation, data processing, and software and data distribution. The partners are interested in creating federations of related data. The federations will serve a plurality of user needs by capturing the different characteristics and properties of data. To help users understand data in the various databases, they require a high-level conceptual representation so that users can determine what queries the data supports.

J. Cushing: "Computational Proxies: Modeling Scientific Applications in Object Databases."

Running computational-chemistry experiments is enormously complex. This complexity produces a steep learning curve. Experiments are computationally intensive and long-lived (days, weeks, months); they involve complex data management tasks (intermediate data, many files to name and manage); applications are not interoperable; and experiments require heterogeneous computing resources. To address this complexity, Cushing is exploring how database solutions can help, such as by maintaining information about past runs to help set up new experiments. The computational proxy environment also helps by launching and controlling applications from definitions in a database. She is exploring whether this approach can generalize to other applications, such as in biology.

Peter Li: "The New GDB: A Federation Experiment"

GDB suffers many problems because it is a monolithic database. For example, some classes of users require access to only a subset of the database. Therefore, GDB is being redesigned as a federation: different classes of objects will be split out into independent database servers. A new front end will also be designed to allow queries across the servers. The new front end will also be designed to allow queries across the servers, and will be generic to allow independent modification of the servers without necessitating changes to the front end. OPM is being used in the redesign. For information about GDB6.0, the URL is `http://gdbwww.gdb.org:1056`.

Z. Cui: "A Framework for Building Intelligent Applications"

Cui discussed the use of an object-oriented, deductive database for implementing a genetics database. Currently there is no language to describe application requirements, hence the impetus for SLOT – a Specification Language for Object Theories. SLOT draws from KADS (a methodology for knowledge engineering) as well as (ML)2, and adds certain object oriented features–for example, organizing theories into a hierarchy, and allowing message passing between object theories.

Chapter 3

Working Group Reports

3.1 Working Group on Requirements for Inter-Database Analysis and Complex Queries

Members: Mary Berlyn, Judy Blake, Terry Gaasterland,[1] Kate Hearne, Elizabeth Kutter, Dhiraj Pathak, Mary Polacco, Junko Shimura, Dong-Guk Shin, Randy Smith, Lincoln Stein[1], Hideaki Sugawara.

3.1.1 Current Problems In Database Interconnection

- The community databases are themselves imperfect: information is incomplete. For example, there are no online services for providing synteny, taxonomy, anatomy/histology information or integrated genetic maps. Information is inconsistent both within and between databases. Information is out of date. Information is redundant. Some data, such as free text, are inaccessible to queries. In addition, there are a wealth of data embedded in specialized databases that are not accessible through existing interdatabase links.
- Mechanisms for community-based curation, annotation, and updating are weak.
- The lack of a common data exchange format requires data to be reformatted when exporting to analysis and reformatted again when importing analysis results.
- Links between databases are spotty .
- There is no comprehensive and up-to-date list of molecular-biology databases.
- Current systems that allow cross-database queries tend to hide the source of the data and the mechanisms used to reach it.
- The lack of shared data models, or at least of controlled vocabularies, hinders the formation of cross-database links.

[1]This report, written by Terry Gaasterland and Lincoln Stein, reflects the discussions and deliberations of the working group ensemble.

3.1.2 Example Interdatabase Queries

The Working Group felt it would be useful to consider a number of scenarios that cannot be answered using currently existing community databases. This is not an attempt to examine an exhaustive list of query classes. Instead it is an attempt to demonstrate that simple hyperlink-based data browsing does not suffice for many classes of realistic biology exploration.

- If I interrupt serine biosynthesis (pathway or step) what is the range of phenotypes I might observe based on:
  1. Information from pathway disruption
  2. Information from metazoans
  3. Information from angiosperms
- Given a putative disease locus partially mapped between two known genetic markers:
  1. What are the genes that have been mapped in this region in the human?
  2. What are the genes that have been mapped in this region in nonhuman organisms (using synteny information)?
  3. Give the sequence ids of the genes.
  4. What is the known function of each gene that has been mapped in those regions?
- Return all human gene sequences with functional (EC#) annotation for which a nonvertebrate homologue exists. Include annotations and map coordinates. (DOE Query #4)

The following graphs represent two alternative "data flow" pathways for Q3. In graph 1, the query asks for human genes that are connected to invertebrate genes which are in turn connected to EC numbers. Then for each of those human genes, get the map coordinates. In graph 2, the connections to nonvertebrates and to map coordinates occur simultaneously with the intersection as the result.

```
Graph 1:

HUMAN GENES <--  INVERTEBRATE GENES <-- EC#
       |
       |
Accession# + annotation <-- MAP COORDINATES
       |
       |
Accession# + annotation + map coordinates




Graph 2:

EC# --> NONVERTEBRATE -->  HUMAN <-- MAP COORDINATES
```

```
   GENES    GENES
                              |
                              |
Acc# + annotation + map coordinates
```

Evaluating this query manually requires

1. Either links or annotations, maps, and EC#s in local database
2. Traversing links between multiple databases
   - Knowing which databases have which information
   - Merging responses from each database
3. A query language that combines schema definition and location of information

3.1.3 Suggested Database Tools

- The Working Group expressed a strong disinclination toward interdatabase query tools that hide the mechanism used to arrive at the result. Different databases have different subjective reliabilities and the user wishes to know the methods by which the answer was derived, in analogy to the materials and methods used to derive the results of an experiment.

  The preferred tool would be one allowing the user to construct the query interactively, giving him or her as much or as little control over the details as desired.

  Desirable features include

  - The ability to control routing of subqueries to different databases
  - Offline query construction, allowing the query to be examined and modified before it is submitted
  - Query by example
  - The ability to incorporate data analysis and data manipulation tools in the query
  - Access to one or more controlled vocabularies
  - User constraints
  - Graphical and natural language extensions.

- The Working Group felt it was vital that the answers returned from queries be structured so that the user has the option to selectively review the source and attributions of each part of the response.

- The Working Group felt that there is a need for community curation tools that

  - Enable individuals to correct mistakes in community databases
  - Enable individuals to enrich community databases by adding interdatabase links and searchable annotations
  - Record the identity of the individual making each modification

- The Working Group felt that related data analysis tools should share a common data format, and that the query tools be able to retrieve data in these formats. In addition, it should be possible to incorporate the output of these tools directly into the databases as annotations.

- The Working Group felt there was a need for structured data browsing tools. Such tools would provide higher-level concept-based browsing that would complement the current hyperlink-based browsing tools.

3.1.4 Priorities and Recommendations

- Establish a curated and up-to-date database of databases, possibly to be used as a services directory and a source of unique database identifiers.
- Encourage community curation and quality control by developing funding and citation mechanisms to reward database curators and by greatly simplifying the process of making corrections and annotations.
- Accelerate the incorporation of interdatabase links. The Working Group suggests that the primary authors be encouraged to provide links when known, and that databases provide mechanisms to allow the community to add annotations and links to specialized databases.
- Develop interactive query tools for interdatabase queries.
- Integrate data analysis tools with query and curatorial tools by encouraging the adoption of a common data exchange format.

3.2 Working Group on Schema Semantic Documentation

Members: Yigal Arens, Matthew Corey Brown, Philipp Bucher, Judy Bayard Cushing, Susan Davidson, David George, Tim Littlejohn, Chris Overton, Owen White.

3.2.1 Goal

The goal identified by the working group was to characterize the types of information that people find useful when trying to understand and use a database, as a basis for a more formal approach to semantic schema documentation. This information should be sufficient to support the following tasks:

- Database integrators should be able to understand the relationship of the database to other databases, such as being able to identify semantic mismatches between representations of similar entities in this database and other databases.
- Potential users should be able to tell whether the database will be useful for them
- Potential users should be able to discover how to use the database.
- Database administrators should know how to continue maintaining the database.

Semantic information about the database may be provided at any *level*–that is, it may describe the database as a whole, objects/blobs in it, views, attributes, and so forth. This approach will support a leveled understanding of the database, providing the users with a high-level overall description of the

database, down to a more detailed understanding of specific components, as appropriate. The types of information that seem to be helpful to users when trying to decide if a database is potentially useful include high-level English descriptions of the various components, an understanding of the source of the information, an understanding of how and when it is updated, and an understanding of how the data is controlled or verified. In addition, extensive examples are immensely helpful. Our reason for suggesting that the information be divided into the categories that appear below is twofold:

1. To provide for clear placement for different types of information
2. To allow changes in some aspects of the database's contents and organization (and hence its description) without affecting others

The types of information that must be provided are

1. Conceptual: A characterization of the kind of objects included in each database class. Necessary and sufficient conditions for inclusion of an instance in the class.
2. Policy: How are candidates for addition to or modification of the database obtained? What sources are searched? What procedure does an outside contributor follow?
3. Verification: A statement of the methods for enforcement of the policy.
4. Logical: The "standard" description of the database, including data model and data dictionary. Along an orthogonal dimension, the derivation and freshness of data is important. To capture this, objects at any levels will be characterized as
    - **Primary** — generated independently by the authors of this database
    - **Derivational** — derived by the processing and/or manipulation of data from other sources, which must be identified
    - **Linked** — copied from another identified source
    - **Heterogeneous**— an unspecified combination of the above

In addition, objects may be time-stamped with their inclusion or last update time. The level of granularity at which such descriptions are provided will be, among other things, a function of the resources available to the database administrator. The more specific the information, the more likely it is to be helpful to potential users. The different types of semantic information (e.g., conceptual, policy) will make reference to the derivational/time-stamp information as appropriate.

3.2.2 A Simple Example

The description of a citation database might include the following:

- Object: Citation.
- Concept: A list of unique references, each of which is to a body of work reporting a sequence.

- Policy: On the first of every month a specific set of journals are scanned for new citations; citations are considered equivalent when they purport to report the same sequence.
- Verification: We verify that citations are reporting the same sequence by (1) checking that their authors and institutions are the same, or alternatively that the accession numbers are the same and (2) checking that the sequences reported are the same: identity is determined by direct string comparison.
- Logical: The data are represented as a table in a Sybase database, with columns Author, Institution, Accession Number, ...

Note that a variety of formats could be used to convey this information. In particular, those describing the database should be encouraged to provide example database entries.

3.3 Data Definition Language Working Group

The working group heard the following presentations on specific data models:

- Victor Markowitz on the Object-Protocol Model (OPM)
- Stan Letovsky on Genera
- Richard Durbin on AceDB
- John Gennari on Ontolingua

The working group also considered the issues involved in establishing a "common data model" and certain problems encountered in translating among data models. The terminology in this field is varied and confusing with many "almost equivalent" terms in common use. Key definitions are as follows:

- A data model is a formalism for defining the structure and other properties of data elements. An almost equivalent term is type system. An example is the relational data model.
- A data definition language is a formal language for expressing the definitions of data elements. An example is SQL. Usually, data models and data definition languages are intellectually linked, because one needs a data definition language to use a data model in practice. In principle, multiple data definition languages could be defined for a given data model, though this is rarely done in practice.
- A schema is the definition of one or more related data elements in some application domain (such as biology) expressed in a specific data definition language. Almost equivalent terms are type definition, class definition, record type, model (as used in AceDB), and ontology.
- A database is a collection of data elements (instances) conforming to a schema.

3.3.1 Specific Data Models

All of the data models, except Ontolingua, are domain-specific and are targeted at molecular biology. These models share a core of data-modeling concepts that may be loosely characterized as semantic data models, or perhaps structural object-oriented data models. The basic data elements that can be defined are nested record structures with some notion of primary key or object identifier. The fields or attributes of these records can be single- or multi-valued and may permit or prohibit NULLs; the models differ as to which of these choices are the defaults. Field values may be

- Built-in types, such as numbers or strings
- Constructed types, such as tuples, sets, lists, nested structures, or references to other objects

The models differ in the precise type constructor provided–for example, AceDB has no explicit list constructor.

The data definition languages of these models are quite different at the syntactic level. OPM employs punctuation marks such as parentheses and brackets in a manner that is familiar to software developers, but which may appear formidable to biologists. Both AceDB and Genera use line-oriented syntax in which line boundaries and other "white space" are syntactically significant; the resulting language is thought to be more natural for biologists but violates established language design principles in the computer field and may be more error-prone. Some participants expressed the view that these syntactic differences are, from a technical standpoint, relatively minor, and that it should be possible to robustly translate the core modeling concepts from language to language.

None of the models consider procedural object-orientation – that is, the models do not allow objects to include methods. The models differ significantly in the area of inheritance or subtyping.

In addition to the above core data modeling concepts, OPM includes specialized constructs for defining laboratory protocols.

Genera is designed for use as a modeling layer on top of a relational database management system; OPM is more general because it is DBMS-independent, although it is currently implemented on top of a relational DBMS. The AceDB data model is designed for use as part of an integrated software system that includes a database management system and a number of user interface displays. The AceDB data model is tightly linked to a data exchange format, called .ace format; any AceDB database can be converted into .ace format, transmitted to another AceDB system, and reconstituted as a database.

Ontolingua is a metadata model that could conceivably be used to define the other models. More information on Ontolingua can be obtained via the WWW at URL `http://www-ksl.stanford.edu/knowledge-sharing/README.html`.

3.3.2 Common Data Model

Given that the domain-specific data models discussed above share a common conceptual core, it is plausible to consider whether these models could be unified into a common data model.

A unified data model of this sort would offer a number of advantages. By encouraging developers to create schemas using a common language, a unified model would allow other developers to incorporate

those schemas in their own work. It would also make it easier to share the databases themselves, and for developers to reuse programs that access databases.

Two principal disadvantages were identified. First, the adoption of a common data model would render obsolete schemas and software developed for other data models. Second, there is a concern that standards can stifle innovation. The working group did not reach a consensus on this issue.

Perhaps the conclusions to be drawn here are that (1) there is a cost to interoperation (e.g., the cost of revamping software due to changes in the data model), so we can't expect existing databases to be simply plugged into the Internet and interoperate for free, and (2) funding agencies need to recognize and plan for these costs.

3.3.3 Schema Translators

Without a common data model, software tools for schema translation could facilitate interoperation. Peter Karp presented the idea of translators that interconvert several of the most popular schema languages. Such translators would have several benefits:

- Electronic publishing of database schemas — Users cannot intelligently query a database without understanding its contents (the working group on schema semantic documentation identified the importance of this issue). A tool that automatically converted database schemas into HTML format would allow users to easily browse database schemas on the WWW, helping to solve the documentation problem. This approach would further benefit if comments and other documentation were embedded within the schema.
- Ontology translation — Schema design in biology is extremely time-consuming. A schema translation tool would allow the designers of a new metabolic-pathway database to adopt an existing conceptualization of metabolic pathways, but to implement the database using a different database management system.
- Generic database interface — Several groups are designing generic database interface tools that are driven by metadata. That metadata must be accepted in a certain format, such as OPM. Schema translators could make such interfaces more widely applicable by translating the schemas of other types of databases (e.g., AceDB) into OPM. Thus, a generic system for querying or updating databases could operate over a wide variety of database types.

Schema translation is not an easy problem. Its difficulty is at least proportional to the differences among data models. Translating between similar data models such as OPM, AceDB, and ASN.1 would be easier than translations between, for example, an object-oriented schema and a relational schema. One difficulty identified in the working group concerns the problem of translating the "intent" of a schema in situations in which the original data model was inadequate to express the intent cleanly.

3.4 Working Group on Problems of Database Interoperation and Integration

This working group was formed to review problems encountered by informatics researchers when attempting to retrieve and integrate data from several molecular-biology data sources. These problems where then divided into three categories and suggestions were made as to how they may be addressed. The suggestions are directed at those groups in the community who are providing public databases and/or software interfaces to these databases.

The categories of problems are

- Linking databases
- Interfaces and services
- Program and data documentation

3.4.1 Linking Databases

Although many talks at the conference focused on restructuring databases, and schema merging, our group looked at the more basic problem of establishing primitive links between entries in different databases. By "links" we essentially mean cross-database references like the common use of an accession number or Medline id outside of the originating database.

The most outstanding problem with links is that some of the most obvious ones are simply not there. This group recognizes that the creation and maintenance of references to external databases is not an easy task. Besides the actual work of constructing a feasible mapping between the databases, organizational and political problems also exist. For this reason, it was suggested that it may be appropriate for third parties to involve themselves in the creation of multidatabase linking tables.

Since the quality of links has a large impact on their effective use, this group suggests that, where possible, links should

- Be bi-directional

  Both data providers and users should provide the links. It was pointed out, however, that allowing groups to work independently without direct contact can be more effective (i.e., the WWW model).

- Be valid, not "stale"

  As database entries and identifiers change, links need updating. It is expected that once more software is available for using the the links, both users and data providers can more easily check the validity of links and take appropriate action.

- Be stable

  In the same way that we require database identifiers to be stable, we would also like links to be stable. It may be possible to use the same techniques for both tasks.

- Have a documented origin

  Knowing how and why data is related is important for proper use of the relationship. The nature and quality of links depends on the techniques used to create them. In the case where two database entries are derived from each other, then the link is very strong and has a precise definition. Probabilistic or computationally derived links have a very different nature, however, and these techniques need to be documented and their results qualified. In addition, it is important that links come with an explanation of possible errors.

- Have high resolution

  We are rapidly approaching a point when it is not enough to say that an entry in one database is (vaguely) related to an entry in another. Effective linking of data, rather, requires that appropriate subcomponents (i.e., features) are described in the linking information.

- Be insulated from schema changes

  This would allow programs to use the links even after the databases have been reorganized. It was pointed out that this requirement can be in conflict with that of having high resolution links because they would use parts of the schema that are more susceptible to change.

3.4.2 Interfaces and Services

Often the limitations of programs through which data is retrieved impede database integration efforts. We have identified the following core problems based on our own experience with existing software used in the community:

- Lack of low-level interfaces for batch processing

  Programs that have been designed for end-user retrieval of data often lack the ability to perform the same tasks in batch mode or do not allow access to lower-level interfaces. Offering a more open architecture with sample queries would allow more flexible retrieval and make it easier to integrate data from several sources.

- Limited expressibility of data queries

  Fixed queries over a certain set of indexed values may be an adequate solution for some users, but it can be very limiting for those who need to do complex manipulations and integration of data. Expressive, ad-hoc query facilities allow rapid and flexible construction of queries and empower users to explore data more thoroughly. Of particular importance is the ability to filter on any component of an entry (selection), retrieve only relevant components of an entry (projection), and retrieve related entries (join).

- Data retrieval mechanisms inextricably tied to application programs

  From a data integration point of view, it is very important that programs for data retrieval can coexist and be compatible with each other. Where possible, data retrieval procedures should be independent of application programs, and so can be used for the extraction of data alone. Ideally, a layered architecture would provide an appropriate separation of these functions and make it easier to embed (and swap) several different data drivers.

- Heterogeneity of data formats

  Using data from several different databases often requires the use of several parsers (i.e., for flat files) or several APIs. A common data exchange format would alleviate this problem and lessen the need to do pairwise translations between all databases.

3.4.3 Documentation

Lack of documentation of either the database or the API used for retrieval can also stand in the way of database interoperation. Problems include

- Lack of documentation for both programmer and end user

  While the the enduser is more interested in how the program works and what kind of data is in the database, the programmer needs to know more about the structure of the data and how it may be meaningfully assembled through queries.

- Lack of comprehensive documentation

  This is a general complaint – it is better to have too much information, and be forced to sift through it, than not to have enough.

- Lack of documentation at multiple levels of abstraction

  At the highest level, it is necessary to explain the concepts and biological entities that are represented in the database. Documentation should start at this point and proceed toward more concrete descriptions of the data.

Appendix A

Program Committee

Workshop Organizer: Peter Karp, SRI International

Dan Davison, U of Houston
David George, NBRF
Chris Overton, U of Pennsylvania
Robert Robbins, Johns Hopkins U
Robert Murphy, Carnegie-Mellon U
Victor Markowitz, LBL
Tim Finin, U of Maryland
Richard Lathrop, MIT
Susan Davidson, U of Pennsylvania
Michel Noordewier, Rutgers U
Chris Fields, TIGR
Xiaolei Qian, SRI International

Appendix B

Workshop Attendees

Yigal Arens
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
email: arens@isi.edu

Emmanuel Barillot
Genethon
13 place de Rungis
75013 Paris
FRANCE
email: Emmanuel.Barillot@genethon.fr

David Benton
National Center for Human Genome Research
Building 38A, Room 610
National Institutes of Health
Bethesda, MD 20892
email: benton@nchgr.nlm.nih.gov

Mary Berlyn
Department of Biology - O.M.L.
Yale University
156 Prospect Street
New Haven, CT 06511
email: mary@cgsc.biology.yale.edu

Judith Blake
The Institute for Genomic Research
932 Clopper Road
Gaithersburg, MD 20878
email: blake@tigr.org

Dr Gerard J.F. Blommestijn
Department of Biophysics
The Netherlands Cancer Institute (NKI)
Plesmanlaan 121
1066 CX Amsterdam
The Netherlands

email: gblom@nki.nl

Anthony Bonner
Department of Computer Science
University of Toronto
10 Kings College Rd
Toronto, Ont, Canada M5S 1A4
email: bonner@db.toronto.edu

Matthew Corey Brown
3138 Overhulse Road, NW #114
Olympia, WA 98502
email: wildfire@elwha.evergreen.edu

Philipp Bucher
Swiss Institute for Experimental Cancer Research (ISREC)
Ch. des boveresses 155
CH-1066 Epalinges s/Lausanne
Switzerland
email: pbucher@isrec-sun1.unil.ch

Zhan Cui
Imperial Cancer Research Fund
ACL
61 Lincoln's Inn Fields
London WC2A 3PX
England, UK
email: cui@acl.lif.icnet.uk

Judith Bayard Cushing
The Evergreen State College
Olympia, WA 98505
email: cushing@cse.ogi.edu

Susan Davidson
Computer and Information Sciences Department
University of Pennsylvania
Philadelphia, PA 19104
email: susan@cis.upenn.edu

Richard Durbin
Sanger Centre
Hinxton Hall
Cambridge CB10 1RQ
England, UK
email: rd@sanger.ac.uk

Dr Thure Etzold
EMBL
Postfach 10.2209
69012 Heidelberg
Germany
email: etzold@embl-heidelberg.de

Terry Gaasterland
Mathematics and Computer Science Division
MCS 221-D227

9700 S. Cass Avenue
Argonne National Laboratory
Argonne, IL 60439
email: gaasterland@mcs.anl.gov

Christine Gaspin
INRA
station de biometrie-Intelligence Artificielle
BP 27
31326 Castanet-Tolosan
France
email: gaspin@toulouse.inra.fr

John Gennari
Section on Medical Informatics
Stanford University
Stanford, CA 94205
email: gennari@camis.stanford.edu

David George
Protein Information Resource
National Biomedical Research Foundation
Georgetown University Medical Center
3900 Reservoir Road, N.W.
Washington, D.C. 20007
email: george@nbrf.georgetown.edu

Nathan Goodman
Whitehead Institute
Center for Genome Research
One Kendall Square, Building 300
Cambridge, MA 02139
email: nat@genome.wi.mit.edu

Mark Graves
Department of Cell Biology
Baylor College of Medicine
One Baylor Plaza
Houston, TX 77030
email: mgraves@bcm.tmc.edu

Kyle Hart
University of Pennsylvania
510 Blockley Hall
523 Guardian Drive
Phila, PA 19104
email: khart@cbil.humgen.upenn.edu

Catherine Hearne
Advanced Computation
Imperial Cancer Research Fund
61 Lincoln's Inn Fields
London WC2A 3PX
England, UK
email: ch@acl.lif.icnet.uk

Carsten Helgesen

Department of Informatics
University of Bergen
Hoyteknologisenteret
5020 Bergen
Norway
email: carstenh@ii.uib.no

Marie-Francoise Jourjon
INRA/station de biometrie-Intelligence Artificielle
BP 27
31326 Castanet-Tolosan
France
email: mfj@toulouse.inra.fr

Peter D. Karp
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
email: pkarp@ai.sri.com

Wolfgang Klas
GMD-IPSI
Dolivostr. 15
D-64293 Darmstadt
Germany
email: klas@darmstadt.gmd.de

Anthony Kosky
Dept of Computer and Information Sciences
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104
email: kosky@saul.cis.upenn.edu

Elizabeth Kutter
Biophysics
Science, Technology, and Health
The Evergreen State College
Olympia, WA 98505
email: t4phage@u.washington.edu
kutterb@elwha.evergreen.edu

Stan Letovsky
Genome Data Base
Johns Hopkins University
2024 E. Monument Street
Baltimore, MD 21205
email: letovsky@gdb.org

Peter Li
Genome Data Base
Johns Hopkins University
2024 E. Monument Street
Baltimore, MD 21205
email: peterli@gdb.org

Tim Littlejohn
Departement de biochimie
Universite de Montreal
C.P. 6128, Centre-ville
Montreal (Quebec), H3C 3J7
CANADA
email: tim@bch.umontreal.ca

Victor Markowitz
Lawrence Berkeley Laboratory
Mail Stop 50B-3238
1 Cyclotron Road
Berkeley, CA 94720
email: vmmarkowitz@lbl.gov

Subhasish Mazumdar
Computer Science Department
New Mexico Tech
Socorro, NM 87801
email: mazumdar@cs.nmt.edu

John McCarthy
Lawrence Berkeley Laboratory
Mail Stop 50B-3238
1 Cyclotron Road
Berkeley, CA 94720
email: jlmccarthy@lbl.gov

Rachel Oberai-Soltz
Digital Equipment Corporation
50 Nagog park AK02-2/D8
Acton, MA 01720
email: roberai@akocoa.enet.dec.com

Dr. G. Christian Overton
Department of Genetics
University of Pennsylvania School of Medicine
CRB 476
415 Curie Blvd
Philadelphia, PA 19104-6145
email: coverton@cbil.humgen.upenn.edu

Suzanne Paley
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
email: paley@ai.sri.com

Dhiraj K. Pathak
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213
email: dkp@cs.cmu.edu

Fabien Petel
Department of Genetics

School of Medicine
Stanford University
Stanford, CA 94305-5120
email: fabien@genome.stanford.edu

Mary Polacco
Department of Agronomy
University of Missouri
210 Curtis Hall
Columbia, MO 65211
email: maryp@teosinte.agron.missouri.edu

Xiaolei Qian
Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
email: qian@csl.sri.com

Otto Ritter
German Cancer Research Center (DKFZ)
Department of Molecular Biophysics
Im Neuenheimer Feld 280
D-69009 Heidelberg
Germany
email: O.Ritter@dkfz-heidelberg.de

Robert Robbins
Department of Energy
ER-72 GTN
Washington, D.C. 20585
email: robbins@er.doe.gov

Patricia Rodriguez-Tome
Genethon
13 place de Rungis
75013 Paris
FRANCE
email: Patricia.Rodriguez-Tome@genethon.fr

Claude Scarpelli
Genethon
13 place de Rungis
75013 Paris
FRANCE
email: claude.scarpelli@genethon.fr

Junko Shimura
Life Science Research Information Section
Institute of Physical and Chemical Research
2-1 Hirosawa, Wako-shi
Saitama 351-01
Japan
email: junko@viola.riken.go.jp

Dong-Guk Shin
Computer Science and Engineering

260 Glenbrook Road
University of Connecticut
Storrs, CT 06269-3155
email: shin@cse.uconn.edu

Tom Slezak
Human Genome Center
Lawrence Livermore National Labs
7000 East Avenue, L-452
Livermore, CA 94550
email: slezak@llnl.gov

Randall F. Smith
Department of Molecular and Human Genetics, T921
Baylor College of Medicine
Houston, TX 77030
email: rsmith@bcm.tmc.edu

Erik Sonnhammer
Sanger Centre
Hinxton Hall
Cambridge CB10 1RQ
England
email: esr@sanger.ac.uk

Lincoln Stein
Genome Center
Massachusetts Institute of Technology
Building 300
1 Kendall Square
Cambridge, MA 02139
email: lstein@genome.wi.mit.edu

Hideaki Sugawara
WFCC World Data Center on Microorganisms
The Institute of Physical and Chemical Research(RIKEN)
2-1 Hirosawa
Wako-shi
Saitama 351-01
JAPAN
email: sugawara@viola.riken.go.jp

Hidetoshi Tanaka
ICOT
1-4-28 Mita
Minato-ku
Tokyo 108
Japan
email: htanaka@icot.or.jp

Dominique Tessier
INRA
rue de la Geraudiere
BP 527
44026 Nantes Cedex
France
email: tessier@nantes.inra.fr

Richard J. Waldinger
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
email: waldinger@ai.sri.com

Owen White
The Institute for Genomic Research
932 Clopper Road
Gaithersburg, MD 20878
email: owhite@tigr.org

Gio Wiederhold
Computer Science Department
Stanford University
Stanford, CA 94305
email: gio@cs.stanford.edu

Manfred Zorn
Lawrence Berkeley Laboratory
MS 50B-3238
1 Cyclotron Road
Berkeley, CA 94720
email: mdzorn@lbl.gov

Appendix C

Workshop Schedule

Tuesday, August 9

9am Session

Welcoming Remarks
P. Karp, SRI International

Overview of Molecular-Biology Databases
R. Smith, Baylor College of Medicine

10:15am Break

10:30am Session: Survey of Computer-Science Approaches
to Database Interoperation

S. Davidson, U Pennsylvania
X. Qian, SRI International

12:00pm Lunch

1:30pm Session: Overview of Working-Group Projects

P. Karp, SRI International

3:15pm Session: Requirements for Database Interoperation

G. Wiederhold, Stanford University

3:45pm Break

R. Robbins, Department of Energy

T. Slezak, Lawrence Livermore Laboratory

M. Graves, Baylor College of Medicine

6:00pm Dinner in the Stanford Rhodin Garden

Wednesday, August 10

8am Breakfast

9am Working Group Sessions

12:00pm Lunch

1:30pm Session: Database Interoperation

Demand-Driven Database Integration for Biomolecular Applications
W. Klas, GMD

Y. Arens, USC/Information Sciences Institute

The IGD Approach to the Interconnection of Genomic Databases
O. Ritter, DFKZ

Using a Query Language to Integrate Biological Data
C. Overton, K. Hart, U Pennsylvania

3:45 break

Dynamic Links between AceDB and Molecular Biology Databases
R. Durbin, Sanger Centre

The Case for Componentry in Genome Information Systems
N. Goodman, L. Stein, Whitehead Institute

Schema Interconversion and Publishing
P. Karp, SRI International

5:00pm Depart for Banquet at Mariani Vineyards in Saratoga

Thursday, August 11

8am Breakfast

9am Working Group Sessions

12:00pm Lunch

1:30pm Session: Data-Definition Tools

V. Markowitz, Lawrence Berkeley Laboratory

J. Gennari, Stanford University

2:30pm Session: Databases

The IDB Database System and its Use in the Human Genome Project: HUGEMAP
E. Barillot, Genethon

The Eukaryotic Promoter Database EPD and its Relationship
to the EMBL Nucleotide Sequence Data Library
P. Bucher, Swiss Institute for Experimental Cancer Research

Session: Database Interoperation

An Integrative Query System for Molecular Biology Databases
T. Etzold, EMBL

3:45 break

J. Blake, O. White, TIGR

Designing an Interface Capable of Accessing Heterogeneous Autonomous
Genome Databases
D. Shin, U Connecticut/Johns Hopkins U

Database Integration using World-Wide Web
S. Letovsky, M. Berlyn, Johns Hopkins U, Yale U

TSL: A Transformation Language for Integration and Evolution of
Biological Databases
S. Davidson, A. Kosky, U Pennsylvania

6:00pm Dinner at Murray House

8:30pm Poster Session, Treat House Dining Room

Friday, August 12

8am Breakfast

9am Session: Database Interoperation

M. Zorn, Lawrence Berkeley Laboratory

Conceptual Design of Macromolecular Sequence Databases
D. George, NBRF

Computational Proxies: Modeling Scientific Applications in Object Databases
J. Cushing, B. Kutter, Oregon Graduate Institute and Evergreen College

P. Li, Johns Hopkins U

A Framework for Building Intelligent Applications
Z. Cui

12:00pm Lunch

1:30pm Session: Database Interoperation

R. Waldinger, SRI International

2:00pm Session: Working Group Reports

3:45 break

4:00pm Session: Future Directions

5:00pm Workshop Ends