INTEGRATING BIOLOGICAL DATABASES

Lincoln D. Stein

Recent years have seen an explosion in the amount of available biological data. More and more genomes are being sequenced and annotated, and protein and gene interaction data are accumulating. Biological databases have been invaluable for managing these data and for making them accessible. Depending on the data that they contain, the databases fulfil different functions. But, although they are architecturally similar, so far their integration has proved problematic.

Over the past two decades, databases of biological knowledge have grown from a cottage industry that was only of interest to a few specialized disciplines, to become essential resources that are used daily by biologists around the world. Examples abound, and include such diverse databases as: PubMed[1], the searchable compendium of biological literature that is maintained by the National Center for Biotechnology Information (NCBI); Ensembl[2], the database of human gene predictions that is maintained by the European Bioinformatics Institute (EBI) and the Wellcome Trust; the UCSC Genome Browser a human, mouse and rat genome browser[3] that is maintained by David Haussler's group at the University of California at Santa Cruz; FlyBase[4], the *Drosophila* research community database that is maintained by the FlyBase Consortium; WormBase[5], the *Caenorhabditis elegans* model-organism database; and the Gene Ontology (GO) database[6] of gene function, process and location terms. Many readers of this article will find it difficult to imagine conducting their work without access to one or more of these databases.

Despite having highly different functions, these databases are all architecturally similar. Each consists of three tiers of software (FIG. 1). At the bottom is a database management system (DBMS) that manages a collection of facts. At the top is the web browser that transmits requests for data to the database and renders the responses as web pages. In the middle is a software layer that mediates between the DBMS and the web browser to turn data requests into database queries, and to transform the query responses into hypertext mark-up language (HTML).

For the biological researcher, however, there are profound differences among the various biological databases. The differences begin on the first page, on which the researcher is greeted by a distinctive look and feel. For example, although Ensembl, FlyBase and the UCSC Genome Browser all provide the similar function of identifying the position of a gene of interest on the human or fly genomes, they provide distinctly different user interfaces for accessing this information. In Ensembl (FIG. 2), the user first selects the 'Human' database, which leads to a search page. Selecting 'Gene' from a pull-down search menu, and entering the name of the desired gene, leads to an intermediate page with a list of genes that have description lines containing the gene name. From here, the user selects the best match, which leads finally to a gene detail page. The position of the gene is printed at the top of this page.

In FlyBase (FIG. 3), the user selects 'Search Genes' from the list of search links on the front page, and then chooses 'Symbol/synonym/name' when prompted for the field to search from. This leads to a table of matching gene symbols, which includes the cytogenetic map position of each gene. Selecting the cytogenetic map position takes the user to a graphical display that shows the position of the gene in base-pair coordinates.

The UCSC browser (FIG. 4) requires the user to select 'Human' from a pull-down search menu and then enter the name of the gene into a search field. This leads to a page that summarizes all matches to the gene name, and, conveniently, lists the gene position directly.

Cold Spring Harbor Laboratory, 1 Bungtown Road, Cold Spring Harbor, New York 11724, USA.
e-mail: lstein@cshl.org
doi:10.1038/nrg1065

ORTHOLOGUE
A homologous gene that is derived from a speciation event or by vertical descent.

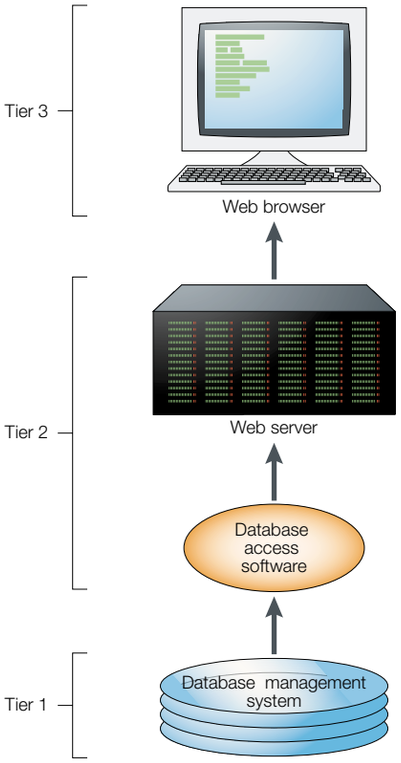

Figure 1 | **Biological database architecture.** Most biological databases use a three-tier architecture that consists of a database management system, a middleware layer and a web interface.

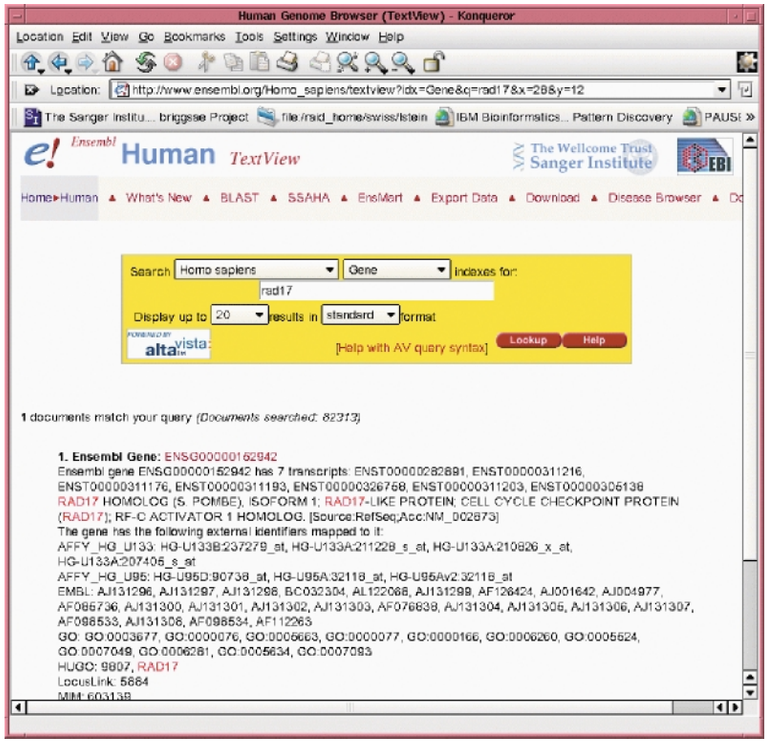

Figure 2 | **Searching for a gene in Ensembl.**

However, the differences are deeper than the user-interface issues. Consider this scenario. A human geneticist has localized an obesity factor to a 5-Mb region of human chromosome 1. Are there any genes in this region that have homologues that are involved in the regulation of lipid metabolism in any model system? To answer this question, it is necessary to traverse several databases. First, the researcher goes to the UCSC or Ensembl databases to find all the predicted genes in the region. Next, the BLAST search engine at NCBI is visited, to find homologues in the various model-organism systems. Then, the researcher goes to the GO web site to look up the GO terms that are related to lipid metabolism. Finally, the researcher either visits each of the model-organism databases in turn to find out whether any of the homologues are related to one of the GO terms, or downloads the entire list of genes that are related to lipid metabolism from the GO web site and checks whether one or more of the homologues are on this list.

The problem is that each of these database resources contains a different subset of biological knowledge. Although each database can answer questions in its domain, it cannot help with questions that span domain boundaries. For example, the FlyBase database does not know about orthology relationships, The Institute for Genomic Research (TIGR) database can help with orthology but not with map position and the UCSC database knows nothing about the fly. As this example shows, researchers must become adept at 'database surfing' to answer many reasonable questions. However, an automated form of this strategy, called data mining, has been raised to a high art in the bioinformatics world.

Integration is difficult

Life would be much simpler if there was a single biological database, but this would be a poor solution. The diverse databases reflect the expertise and interests of the groups that maintain them. A single database would reflect a series of compromises that would ultimately impoverish the information resources that are available to the scientific community. A better solution would maintain the scientific and political independence of the databases, but allow the information that they contain to be easily integrated to enable cross-database queries. Unfortunately, this is not trivial.

There are many integration challenges. One of the most difficult is the one that might seem the most minor — how do you assign and maintain the correct names of biological objects across databases? For example, consider the DNA-damage checkpoint-pathway gene that is named *Rad24* in *Saccharomyces cerevisiae* (budding yeast). *Saccharomyces pombe* (fission yeast) also has a gene named *rad24* that is involved in the checkpoint pathway, but it is not the ORTHOLOGUE of the *S. cerevisiae Rad24*. Instead, the correct *S. pombe* orthologue is *rad17*, which is not to be confused with the similarly named *Rad17* gene in *S. cerevisiae*. Meanwhile, the human checkpoint-pathway genes are sometimes named after the *S. cerevisiae* orthologues, sometimes after the *S. pombe* orthologues, and sometimes have

independently derived names. In *C. elegans*, there are a series of *rad* genes, none of which is orthologous to *S. cerevisiae Rad17*. The closest *C. elegans* match to *Rad17* is , in fact, a DNA-repair gene named *mrt-2*.

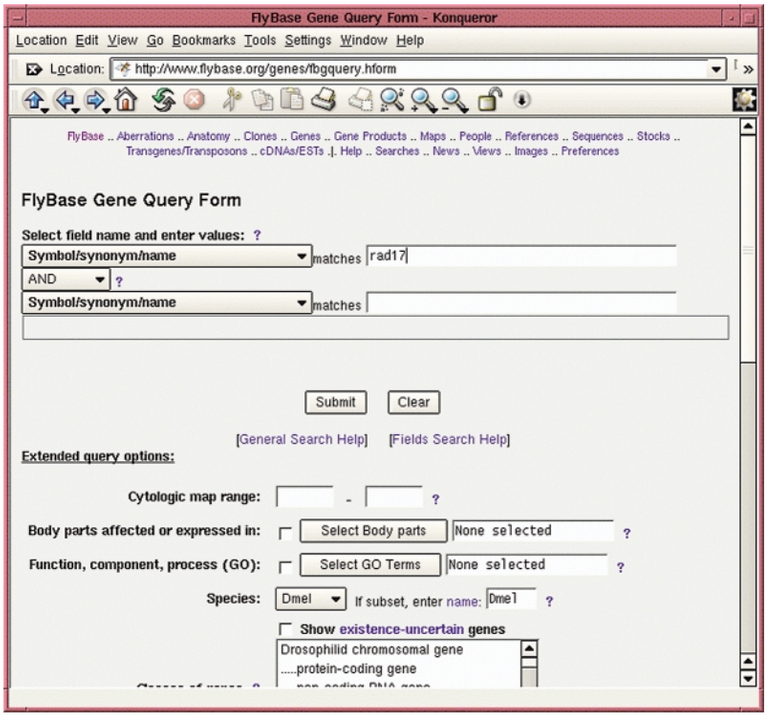

Figure 3 | **Searching for a gene in FlyBase.**

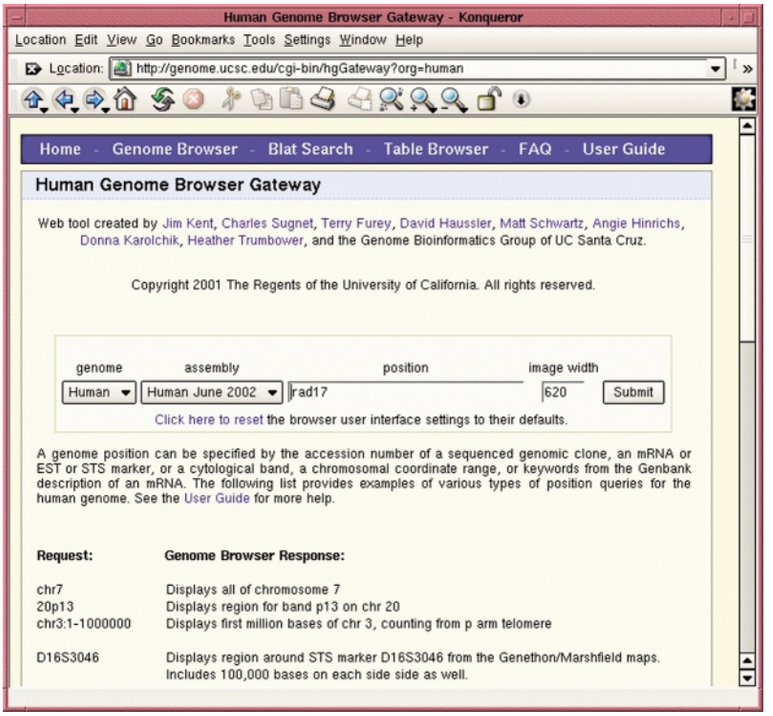

Figure 4 | **Searching for a gene in the UCSC Genome Browser.**

A more subtle problem is the clash of concepts as users move from one database to another. An extreme example, first noted by Michael Ashburner, considers the use of the term 'pseudogene' by different researchers and research communities. To some, a pseudogene is a gene-like structure that contains in-frame stop codons or evidence of reverse transcription. To others, the definition of a pseudogene is expanded to include gene structures that contain full open reading frames (ORFs) but are not transcribed. Some members of the *Neisseria gonorrhea* research community, meanwhile, use pseudogene to mean a transposable cassette that is rearranged in the course of antigenic variation.

There are also more subtle disagreements. The human genetics community uses the term allele to refer to any genomic variant, including silent nucleotide polymorphisms that lie outside of genes, whereas members of many model-organism communities prefer to reserve the term allele to refer to variants that change genes. Even the concept of the gene itself can mean radically different things to different research communities. Some researchers treat the gene as the transcriptional unit itself, whereas others extend this definition to include up- and downstream regulatory elements, and still others use the classical definitions of cistron and genetic complementation.

There are also technical challenges. The various biological databases use different DBMSs and none provide a standard way of accessing the data. Some databases provide large text dumps of their contents, others offer access to the underlying DBMS and still others provide only web pages as their primary mode of access.

Even more challenging is the issue of updates — biological databases are always changing, so integration must be an ongoing task.

Integration approaches

There are three main ways in which groups have tried to integrate biological databases, which are referred to here as link integration, view integration and data warehouses.

Link integration. Link integration has been by far the most successful, because it lends itself to the haphazard nature of the web. In the familiar and ubiquitous version of this technique, researchers begin their query with one data source, and then follow hypertext links to related information in other data sources. For this to work well, the data sources must cooperate to create dependable linking rules, but this is not a strict requirement and many web-based databases link to external databases without the knowledge or cooperation of the managers of the external site.

A variation on link integration is embodied in the popular sequence retrieval system (SRS)[7], which is a keyword indexing and search system for biological databases. SRS is more sophisticated than general web-based search tools — for example, Google — because it recognizes the existence of structured fields in source databases and allows maintainers to explicitly relate a field in one database to a differently named field in another.

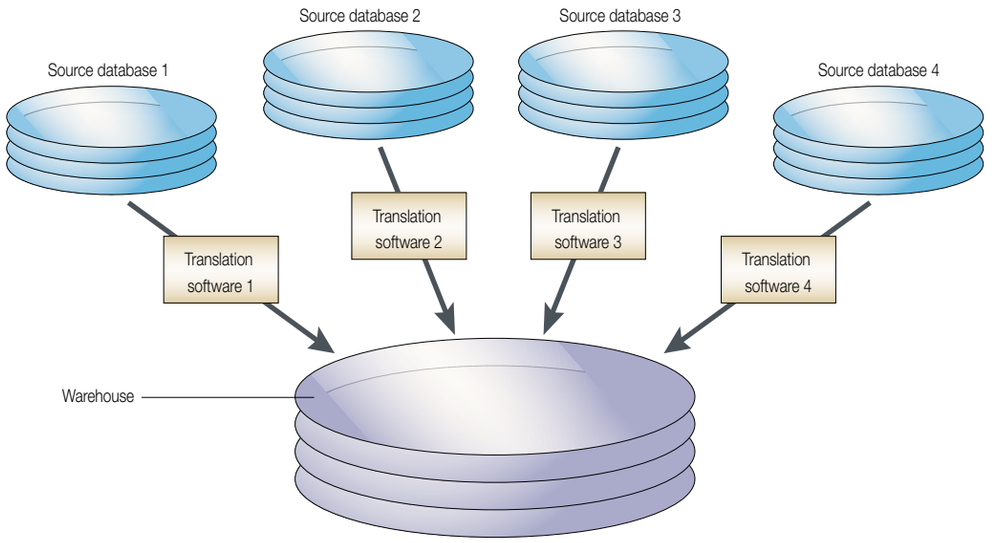

Figure 5 | **Data warehousing.** The data warehouse technique transforms the contents of multiple source databases to fit a common data model. It then integrates the source data into a single large database.

Unfortunately, link integration is problematic on several fronts. First, it is extraordinarily vulnerable to naming clashes and ambiguities. For example, a naive researcher who tries to navigate through the *Rad24* gene confusion using SRS or direct database-to-database linking might wander into the wrong gene family. Second, there are update issues. An outgoing web link represents a leap of faith that the page at the other end of the link is still valid — if it exists at all. For example, if the curators of the *S. cerevisiae* database withdraw or rename *Rad24*, they have no way of informing all the databases that point to the *Rad24* page that an update is required. Third, as we are all too familiar with on a daily basis, link-level integration puts the onus of integration and interpretation on the researcher.

View integration. View integration leaves the information in its source databases, but builds an environment around the databases that makes them all seem to be part of one large system. The most complete attempt at this involved the development of the cross-database query languages Kleisli and K2 at the University of Pennsylvania in the 1990s (REF. 8). If either of these languages is given a query, the language processor analyses the query to discover which databases need to be accessed to satisfy the request, and generates a series of subqueries. The processor then hands the subqueries several 'drivers' that can extract the information from particular databases — for example, the GenBank driver, which can query the NCBI Entrez web interface. After the drivers fetch the data, the Kleisli/K2 query processor transforms and integrates it, and returns the data to the user.

Despite the appeal of this approach, the system has failed to catch on with the community. Researchers might be disappointed in the performance of the system. Because processing a query is limited by the slowest data source, Kleisli and K2 rarely have the performance that is associated with direct access to the source databases. The failure of these languages to be adopted by the academic bioinformatics community is more puzzling, but might result from the complexity of writing and maintaining the component database drivers. Supporting this view is the fact that a few commercial entities are using K2 to manage in-house data — their large bioinformatics groups can handle the complexity of the system, and their control of in-house databases simplifies the maintenance issues.

Data warehousing. The last general approach can be broadly described as bringing all the data under one roof in a single database (FIG. 5). The first step in data warehousing is to develop a unified data model that can accommodate all the information that is contained in the various source databases. The next step is to develop a series of software programs that will fetch the data from the source databases, transform them to match the unified data model and then load them into the warehouse. The warehouse can then be used as a 'one-stop shop' for answering any of the questions that the source databases can handle, as well as those that require integrated knowledge that the individual sources do not have.

It is more difficult to create a data warehouse than it might sound. The single biggest issue is keeping the data warehouse up to date. New information is being continually added to the source databases, which means that the new data must be re-imported into the warehouse in a timely fashion or the warehouse will go out of date. To make matters worse, database designs do not stand still; their maintainers are continually tinkering with the data model by adding new data types, changing fields and nomenclature, and changing the relationships among data types. This constant churn means that dump, transform and load software that have been written for one version of a database will not necessarily work with a later version.

One of the most ambitious attempts at the warehouse approach was the Integrated Genome Database (IGD) project[9], which aimed to combine human sequencing data with the multiple genetic and physical maps that were the main reagent for human genomics at the time. At its peak, IGD integrated more than a dozen source databases, including GenBank, the Genome Database (GDB) and the databases of many human genetic-mapping projects. The integrated database was distributed to end-users complete with a graphical front end.

The IGD project survived for slightly longer than a year before collapsing. The main reason for its collapse, as described by the principal investigator on the project (O. Ritter, personal communication), was the database churn issue. On average, each of the source databases changed its data model twice a year. This meant that the IGD data import system broke down every two weeks and the dumping and transformation programs had to be rewritten — a task that eventually became unmanageable.

A more recent warehouse project that is underway at the University of Pennsylvania makes use of a generalized model for biological data called the Grand Unified Schema (GUS)[10]. The immediate goals of this project are more modest than those of IGD, because the aim is to support several targeted in-house research projects rather than to become a general public resource. It remains to be seen whether GUS-based databases will surmount the scaling problems that brought down IGD.

Web services

An important variant of link-level integration is the concept of the web service. In this view, the heterogeneous collection of loosely-linked data resources on the web is turned on its head, and becomes a world of services that are linked by service names and definitions. In such a world, GenBank is no longer a database from which users retrieve sequence entries, instead it is a service that transforms sequence accession numbers into GenBank FLAT FILES. This concept might seem a minor point, but it allows users to establish a common framework that encompasses, for example, the BLAST search engine, which transforms a search sequence into a series of similarity hit records on a selected query database.

My personal experience with web services comes from my involvement in the development of the Distributed Annotation System (DAS)[11], which was an early example of this genre. DAS provides a web service for exchanging genomic 'annotations' — a term that is used loosely here to indicate anything that can be associated with a region of a genome, such as the position of a predicted gene. The DAS protocol is simple. The user asks for a genomic region of interest by naming the genome (the species and usually the assembly release number), a landmark on the genome (a clone accession number, a genetic marker or a well-known gene name) and a region of the genome relative to that landmark. The server then returns a structured document that contains information about all the annotations that overlap the specified region. The information in this document is basic: the type of annotation, its start and stop positions and an optional uniform resource locator (URL) that can be retrieved to obtain further information about the annotations.

The DAS service allows data providers to exchange information about sequence annotations and allows a limited form of data integration. For example, the Ensembl web server uses DAS to allow third-party data providers to add their own comments to its canonical annotations of genes, repeat families and other genome features. These third-party annotations can be displayed as tracks on the Ensembl sequence viewer, thereby allowing researchers to see how the third-party annotations relate to each other and to the annotations of Ensembl. Another useful feature of DAS is that it allows software developers to write generic programs that read and display DAS-based annotations; for example, both the Apollo genome editor[12] and the Omniview genome viewer provide browsing across arbitrary DAS-compatible databases.

Although DAS has been adopted by a variety of groups, and can be deemed a success, it has several limitations that have become clear with time. One limitation is that it is not easily extended. Groups that wish to create a DAS that works for protein motifs, genetic maps or physical maps cannot easily do so. Neither is it easy to extend DAS to transmit a richer collection of information about annotations than the bare minimum that it allows at present. To do so would risk losing compatibility with applications that only expect the core information to be exchanged.

Another limitation of DAS is that it can be hard to locate DAS servers. There is no way to register a new DAS server or to search the web for one. Consequently, people must depend on bookmarks, link lists and other *ad hoc* techniques for finding servers that might contain useful information.

A more fundamental problem is that DAS is semantically weak. It provides a text structure for transmitting the type and position of genome annotations, but does not control the content of the annotation-type field. So, one sequence analysis group might make an assertion about a series of coding sequence (CDS) features on the genome, whereas another might discuss a set of ORFs — they might, or might not, be talking about the same concept. DAS will faithfully transmit the information, but it is up to the user to determine how to relate the two data sets.

Another reflection of the semantic weakness of DAS is its *laissez faire* attitude toward names. Almost all the biological objects that are exchanged through DAS have names — this includes everything from the name of a gene to the name of a genome. However, there is no control over which names are used. If two different groups place *Rad24* on the genome, are they talking about the same or different genes? Are they even talking about genes at all? And do the different groups agree on what a gene is?

Two technologies, the ontology and the globally-unique identifier, go a long way towards solving these problems.

FLAT FILES
Data files that contain records with no structured relationships.

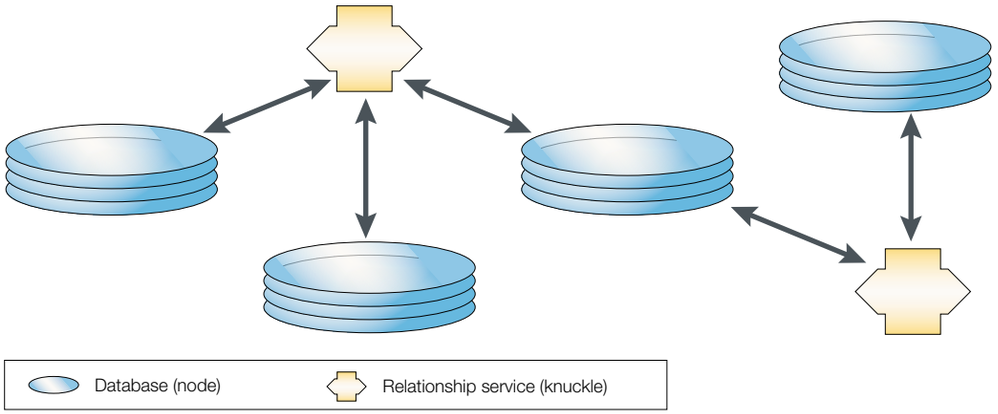

Figure 6 | **Knuckles-and-nodes approach.** In the 'knuckles-and-nodes' approach to integration, the source databases remain independent, but a few important relationships, such as orthologies, are stored in special-purpose linking databases.

The role of ontologies

Ontologies are a sophisticated type of controlled vocabulary that attempt to capture the main concepts in a KNOWLEDGE DOMAIN[13]. I have already referred to the GO that captures information about gene-product location and function in the biological domain. There are also many affiliated biological ontology projects that range from descriptions of mutant phenotypes in plants to anatomical structures in vertebrates (details can be found at the Global Open Biological Ontologies web site).

Although ontologies do not, by themselves, lead to the integration of biological databases, they are important facilitators. The existence of a shared ontology allows an integrator to merge two databases with some guarantee that a term that is used in one database corresponds to the same term used in the other. For example, the sequence ontology (SO) (see the Sequence Ontology Project in online links box) defines a set of terms and definitions that describe features on a genome, such as exon, pseudogene and transcription start site. If two genome databases use SO, it becomes easier to ask integrative questions such as "is there a difference between the average distance from the transcription start site to the translation start site between flies and worms?".

An important feature of biological ontologies is that the terms are organized in a hierarchical manner, so that more specific terms are defined as specializations of more general ones. For example, a portion of the SO contains the following hierarchy: transposable-element insertion, retrotranscribed transposable-element insertion, LINE-class transposable-element insertion, SINE-class transposable-element insertion and retroviral-class transposable-element insertion.

Such a hierarchical approach is advantageous. One genome analysis group might create a database in which the transposable-element insertions are annotated to the detail of the SINE or LINE class, whereas another group might only describe the presence of a transposable element without further subcategorizing it. If both groups use SO terms, then the two databases can be integrated by traversing up the hierarchy tree until the most specific common term is found. In the absence of an organized hierarchy of common concepts, such integration comes down to tedious and error-prone work by hand.

To support the complex relationships that are common in biology, terms are allowed to have more than one parent in a data structure that is formally called a directed acyclic graph (DAG). This makes it possible to express the idea that a retroviral insertion is both a subclass of a transposable-element insertion and a type of repetitive element.

Globally unique identifiers

Shared ontologies can help bioinformaticians agree on how to describe biological objects, but they do not necessarily help them to agree on how to name them. Recall the issues that the same biological object might have multiple names, and the same name might denote multiple objects. One approach is to designate an authoritative names commission to manage the definitive list of such names, as the HUGO Gene Nomenclature Committee is attempting to do with human gene symbols. This rarely works in practice because of the dynamic nature of the field. Whether speaking of genes, proteins, cell lines or strains, names come, go, are merged and split too rapidly for any commission to keep up with. Even if the names commission could handle the name flux, it is unclear how these changes can be propagated to the databases that depend on them.

Although a central naming authority is usually impractical, there are often *de facto* authorities for local subsets of the names. For example, the NCBI can be considered to be authoritative for GenBank identifiers, and WormBase is authoritative for *C. elegans* gene symbols. By simply distinguishing between a name that comes from one locally authoritative source and a name that comes from another, it is possible to effectively eliminate the confusing name clashes. This begs the question of how to establish identity among the multiple synonyms for the same object; nonetheless, it is a good start.

KNOWLEDGE DOMAIN
A body of knowledge that is often associated with a specialized scientific discipline.

LINE
Long interspersed-repeat transposable elements.

SINE
Short interspersed-repeat transposable elements.

There are two main lines of thought among groups that are interested in globally unique identifiers. One line holds that object identifiers should point to the objects themselves and use a URL SYNTAX. To give a specific example, the WormBase database is set up so that all gene objects can be retrieved using a URL. For example, when the page for the *rad-3* gene is fetched in a web browser it returns a description of the *rad-3* gene as an extensible markup language (XML) document. This URL acts as a stable globally unique identifier for the *C. elegans rad-3* gene and cannot clash with objects named *rad-3* that exist in other databases.

One drawback of this scheme is that it couples the identity of a biological object with the location of its representation on the web. If, some day, the WormBase administrators reorganize the site, or if WormBase becomes commercial and replaces its domain name with www.wormbase.com, the identifier for *rad-3* might still be globally unique but the URL will no longer retrieve its contents.

The other way of creating globally unique biological identifiers decouples the notion of the location of a resource from its authoritative source. The Life Sciences Identifier (LSID) proposal, put forward by the Interoperable Informatics Infrastructure Consortium (I3C), combines the internet domain name of the source database with the local identifier from the database. For example, a valid LSID for the *C. elegans rad-3* gene might be urn:lsid:www.wormbase.org:gene/rad-3. The 'urn:' at the start of the LSID identifies the resource as a Universal Resource Name (URN) to distinguish it from a Universal Resource Location (URL). The 'lsid' field identifies the URN as a life-sciences identifier. The following field contains the domain name of the authoritative source. It must be unique for each data source, and is usually, but not necessarily, a resolvable internet domain name. The last field is an arbitrary identifier that WormBase recognizes as a unique name for the gene — in this case, 'gene/rad-3'.

The trade-off with the use of LSIDs is that users can no longer find the object that the identifier points to without help. A companion proposal to the LSID scheme calls for a 'resolver' service that can translate LSIDs into fetchable URLs.

The combination of ontologies and globally unique identifiers increases the chances that web services can exchange data without manual intervention. The next version of DAS will use SO to describe sequence annotation types and LSIDs to identify biological objects. This will make the integration of multiple DAS data sources simpler and less error prone, solving some of the problems from which the original protocol suffered.

Bringing web services to biology

Two projects now seek to extend and generalize the DAS concept beyond its original domain of sharing sequence annotations: the BioMOBY project[14] and the MyGrid project.

BioMOBY. BioMOBY has two central ideas. One is a language for describing biological services in terms of their inputs and outputs. For example, a GO classification service might accept a gene name as its input, and output a set of GO terms that describe the function of that gene. The other is a central registry of services. Bioinformatics services can register themselves in an online database known as 'MOBY Central'. Users (or more typically, application programs that act on behalf of the users) can then search this registry to find services that might be of use.

In theory, this approach would allow for linking requests in a chain: a user interested in what was known about a particular *Drosophila* gene could search MOBY Central for services that take a gene name as an input, and find the gene-name to GO service that is provided by FlyBase. A MOBY-aware software application could then invoke this service, returning a series of GO terms to the user. The user could then ask MOBY Central for all services that accept GO terms and, having found a potential service on the WormBase database, download all the *C. elegans* genes that have similar annotations. Extending this scenario, the user might search MOBY Central for services that accept multiple sequences and return a similarity score, thereby locating a server that offers a sequence alignment service. In this way, the user could relate gene function to sequence similarity between *C. elegans* and *Drosophila melanogaster*.

A limitation of the MOBY design is that it is limited in its ability to describe all aspects of the web services that it supports. As a result, the registry searching facility is at best an imprecise tool. For example, the gene-name to GO term services provided by FlyBase and WormBase are not functionally equivalent because the former will only recognize the accepted names of *Drosophila* genes, whereas the latter acts only on *C. elegans* genes. However, they seem the same to MOBY. Another concern is that a central registry provides a potential 'choke point'. For the system to work, MOBY Central must be accessible 24 hours a day and its operating expenses must be supported indefinitely.

The MyGrid project. The MyGrid project takes a more ambitious approach. It uses information-processing methodologies, which have been developed for the Semantic Web and grid computing[15], to develop precise descriptions of bioinformatics services, and also to describe how they are related, how they can be located and invoked. An important aspect of the MyGrid design is its use of ontologies to describe bioinformatics services themselves. It means that one group could define the bare minimum requirements for a sequence similarity-search service and name it the 'basic similarity-search service'. Another group could later extend this service definition to encompass the thresholds, cut-off values and word-size parameters that are required by the BLAST algorithm and name it the 'BLAST similarity-search service'. The two search-service definitions can then be related to each

SYNTAX
The grammar, structure and order of elements in a language statement. In computing, it refers to the rules that govern the structure of computer commands — for example, statements or other instructions that are used in code.

other by a language that allows a piece of application software to understand that the BLAST service is a special case of a similarity search engine and to invoke it properly.

MyGrid, like MOBY, also aims to support service discovery, but does not necessarily rely on the concept of a central registration system. Under discussion are both registry-style architectures and alternatives that rely on Google-like search engines that discover new services on their own.

Both MyGrid and MOBY use a variety of technologies that have been developed for non-biological web applications — an alphabet soup that includes SOAP/XML, WSDL, UDDI, XSDL, RDF and DAML+ OIL. Reflecting the difference in the scale of their goals, the MOBY project provides demonstration software and toolkits for developers at its web site, whereas MyGrid is at an early conceptual stage.

The knuckles-and-nodes approach

Although it is tempting to treat the integration of biological databases as a technological problem, in fact the main impediment to achieving this goal is not technological but sociological. In the opinion of this author, meaningful scaleable integration cannot be achieved without the cooperation of the data providers[16]. As long as the data providers continue to produce online databases without regard for the way in which the information will be aggregated, integration will be a monumental task. However, in the absence of accepted standards for the represention and exchange of biological data, it is far from simple for data providers to achieve the goal of making their data available in a form that can be cleanly integrated and maintained.

I propose a 'knuckles-and-nodes' approach (FIG. 6). The nodes are source databases, much as we have now, each of which uses a distinct and independent data model and whatever technology suits it. The nodes are rich, biologically detailed and deep. By contrast, the knuckles are carefully maintained and curated services that provide the information that is necessary to relate the data in one database node to data in others. The knuckles are restricted in scope to a single task and are constrained to use a standard interface.

For example, one knuckle could be a service that keeps track of orthology relationships. Such a service would have the responsibility of translating a gene symbol in a particular species into the corresponding symbol(s) in one or more other species. This service would immediately give data providers a stable target for orthology links: instead of linking in a pairwise fashion to the fly, yeast, worm, mouse or human databases, they could simply link to the orthology knuckle. This knuckle could then link out to the appropriate databases for each of the species that has an orthologue. The existence of a definitive reference site for orthologues would also facilitate deeper levels of integration by allowing the species-specific databases to be merged without leading to name clashes and ambiguities, as discussed earlier.

Other knuckles include: a citation knuckle that relates genes and protein products to papers; an EC knuckle that relates proteins to their Enzyme Commission Nomenclature enzymatic activities; a GO knuckle that relates a GO term to its definition and can provide information about the position of the term in the GO relationship graph; a GO association knuckle that relates gene products to GO terms; and a taxonomy knuckle that relates one species to another through the taxonomic tree.

The knuckles-and-nodes design allows individual database maintainers the flexibility to choose their own data model, user interface and DBMS, but retains just enough rigidity at the crucial points so that the contents of one database can be related and integrated with those of another.

For this concept to work, however, several sociological and technological requirements must be met. Sociologically, a 'critical mass' of data providers must meet and agree to create one or more of the knuckles. It is unlikely that a single research group would be entrusted with the responsibility of developing resources such as the definitive orthologue set. Almost certainly, some sort of cooperative agreement would be made in which several groups contribute curated subsets of the information to a common repository and develop a mechanism for adjudicating disagreements.

This idea of cooperation among bioinformatics groups is not as idealistic as it might sound. For example, both the University of Indiana euGenes database[17] and the NCBI LocusLink/RefSeq[1] are nascent attempts to develop definitive catalogues of genes in model species. In both cases, the maintainers of the model-organism databases have been eager to cooperate. Indeed, either euGenes or LocusLink would be ideal starting points for an orthology knuckle. The Gene Ontology Consortium web site already acts as a nexus for information that pertains to gene-product location and function for more than a dozen species.

Technologically, data providers must agree on a common set of software protocols to access knuckle services. In my opinion, the simplicity of the data models and the ease of implementation of knuckles and nodes will make or break such attempts. The more complex a data model, the harder it is for a group of bioinformaticians to reach consensus, but, if a data model is too simple it will be inadequate to address the problem at hand. For this reason, I am pleased to see both complex (MyGrid) and simple (BioMOBY) biological web-service systems on the drawing board. Somewhere between the two poles of the complexity spectrum is the sociological 'sweet spot'.

Integration remains difficult

Despite my overall optimism, the integration of biological data will remain a difficult problem for the conceivable future. Only by a concerted effort on the part of the database providers, and with the encouragement and support of the research community, will we be able to tame the explosion of biological data.

1. Wheeler, D. L. *et al.* Database resources of the National Center for Biotechnology Information: 2002 update. *Nucleic Acids Res.* **30**, 13–16 (2002).
A description of web services for biological databases that use the SOAP software infrastructure.
2. Hubbard, T. *et al.* The Ensembl genome database project. *Nucleic Acids Res.* **30**, 38–41 (2002).
3. Kent, W. J. *et al.* The human genome browser at UCSC. *Genome Res.* **12**, 996–1006 (2002).
4. The FlyBase Consortium. The FlyBase database of the *Drosophila* genome projects and community literature. *Nucleic Acids Res.* **30**, 106–108 (2002).
A highly successful attempt to develop a common controlled vocabulary for describing gene-product function and location.
5. Harris, T. W. *et al.* WormBase: a cross-species database for comparative genomics. *Nucleic Acids Res.* **31**, 133–137 (2003).
6. The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genet.* **25**, 25–29 (2000).
7. Zdobnov, E. M., Lopez, R., Apweiler, R. & Etzold, T. The EBI SRS server — new features. *Bioinformatics* **18**, 1149–1150 (2002).
8. Davidson, S. B. *et al.* K2/Kleisli and GUS: experiments in integrated access to genomic data sources. *IBM Syst. J.* **40** [online] <http://www.research.ibm.com/journal/sj/402/davidson.html> (2001).
This paper describes a web-services approach to sharing genome annotations.
9. Ritter, O., Kocab, P., Senger, M., Wolf, D. & Suhai, S. Prototype implementation of the integrated genomic database. *Comput. Biomed. Res.* **27**, 97–115 (1994).
10 Bahl, A. *et al.* PlasmoDB: the *Plasmodium* genome resource. An integrated database that provides tools for accessing, analysing and mapping expression and sequence data (both finished and unfinished). *Nucleic Acids Res.* **30**, 87–90 (2002).
This paper contrasts efforts by the same group to integrate biological data sources using the federated database and data warehousing approaches.
11. Dowell, R. D., Jokerst, R. M., Day, A., Eddy, S. R. & Stein, L. The distributed annotation system. *BMC Bioinform.* **2**, 7 (2001).
12. Lewis, S. E. *et al.* Apollo: a sequence annotation editor. *Genome Biol.* **3**, R0082.1–R0082.14 (2002).
13. Stevens, R., Goble, C. A. & Bechhofer, S. Ontology-based knowledge representation for bioinformatics. *Brief. Bioinform.* **1**, 398–414 (2000).
A description of web services for biological databases using the CORBA software infrastructure.
14. Wilkinson, M. D. & Links, M. BioMOBY: an open source biological web services proposal. *Brief Bioinform*. **3**, 331–341 (2002).
15. Foster, I. & Kesselman, C. (eds) *The Grid: Blueprint for a New Computing Infrastructure* (Kaufmann, San Francisco, 1999).
16. Stein, L. Creating a bioinformatics nation. *Nature* **417**, 119–120 (2002).
17. Gilberg, D. G. euGenes: a eukaryote genome information system. *Nucleic Acids Res*. **30**, 145–148 (2002).

Acknowledgements

The author thank the anonymous reviewers for their helpful comments, and D. Gessler, M. Wilkinson, N. Goodman and S. Lewis for many illuminating discussions.

Online links

DATABASES

The following terms in this article are linked online to:

LocusLink: http://www.ncbi.nlm.nih.gov/LocusLink
mrt-2 | *rad*
***S. pombe* GeneDB:** http://www.genedb.org/genedb/search.jsp?organism=pombe
rad17 | *rad 24*
***Saccharomyces* Genome Database:** http://genome-www.stanford.edu/Saccharomyces
Rad17 | *Rad24*
WormBase: http://www.wormbase.org
rad-3

FURTHER INFORMATION

BioMOBY project: http://www.biomoby.org
Ensembl: http://www.ensembl.org
FlyBase: http://flybase.bio.indiana.edu
Gene Ontology Consortium: http://www.geneontology.org
Gene Ontology (GO) Database: http://www.godatabase.org/dev/database
Global Open Biological Ontologies: http://www.geneontology.org/doc/gobo.html
HUGO Gene Nomenclature Committee: http://www.gene.ucl.ac.uk/nomenclature
Interoperable Informatics Infrastructure Consortium: http://www.i3c.org
LocusLink: http://www.ncbi.nlm.nih.gov/LocusLink
MyGrid project: http://phoebus.cs.man.ac.uk/twiki/bin/view/Mygrid/WebHome
Omniview genome viewer: http://www.omnigene.org
PubMed: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi
RefSeq: http://www.ncbi.nlm.nih.gov/RefSeq
Semantic Web: http://www.w3.org/2001/sw
Sequence Ontology Project: http://song.sourceforge.net
The Institute for Genomics Research (TIGR) Database: http://www.tigr.org/htdig
UCSC Genome Browser: http://genome.ucsc.edu
University of Indiana euGenes database: http://iubio.bio.indiana.edu
WormBase: http://www.wormbase.org
Access to this interactive links box is free online.