

# DATA SPACE: A DATA WEB FOR THE EXPLORATORY ANALYSIS AND MINING OF DATA

*A data web can simplify the processes of cleaning, transforming, and exploring data in a data mining application. DataSpace is one such mechanism that lets scientists view, retrieve, and apply simple transformations to remote and distributed data.*

**D**ata mining is the semiautomatic extraction of patterns, changes, correlations, associations, anomalies, and other statistically significant structures from large data sets. It differs from statistics because it emphasizes semiautomated processes that are data-driven rather than human-driven.

Data mining consists of four steps:

1. Extracting, cleaning, and transforming data
2. Exploratory data analysis
3. Building a statistical model
4. Deploying a statistical model

For many problems, steps 1 and 2 are the most time-consuming. For example, it is not atypical for a study to spend two or three times longer on steps 1 and 2 than on step 3. Often, the first step acts as a barrier to data exploration. If the work in step 1 is too much, certain data will not be examined at all in a study. As a simple thought experiment, think about how many more documents you can examine with the World Wide Web than when you had to FTP documents and

then open them. Despite the importance of steps 1 and 2, the majority of recent research in data mining has focused on building better statistical models to improve step 3.<sup>1</sup>

There are probably several reasons for this focus. We can analyze and study theoretically algorithms producing statistical models. We can compare algorithms producing statistical models to each other on specific data sets. Of course, this often produces a false sense of comfort in that only data sets showing the new algorithm in a positive light are usually included in experimental studies' published results.

In this article, we describe an infrastructure called DataSpace designed to reduce the time required to accomplish steps 1 and 2. This same infrastructure also facilitates the use of data produced by others and enables the distributed exploration of remote data. DataSpace is an example of a data web—that is, a Web-based infrastructure for working with data. We believe that DataSpace is novel in that it provides a simple mechanism for lowering the cost of extracting, cleaning, transforming, and exploring remote and distributed data. With this type of infrastructure, the data mining of scientific and engineering data becomes significantly easier. Although many tools are available for exploratory data analysis, they are designed to work with local data, not remote or distributed data.

## Data Webs and Grids

By data web,<sup>1</sup> we mean a Web-based infrastructure for data. This article describes a simple data web server and data web client that communicate with a protocol called the DataSpace Transfer Protocol and that supports the following services:

1. Analyzing, exploring, and visualizing remote data with a DSTP client that communicates with a (single) DSTP server. Exploratory data analysis operations on unfamiliar, remote data is a basic advantage of a data web.
2. Merging and transforming distributed data with a DSTP client that communicates with two distributed DSTP servers. As a special case, one DSTP server might be local and another remote. In this case, the role of the merging is to add or append remote data to the local data set in a meaningful way.

Data mining systems generally assume that data is local and import the data into the system from flat files, databases, or proprietary file formats. Once the data is imported, the user can examine it using a variety of statistical and data mining algorithms. Many data systems also include visualization packages.

There are typically two types of distributed data mining systems:<sup>2</sup> those that use agents to move models and those that use agents to move data. The most common type moves models. Local models at each of several distributed sites are built and then moved to a centralized site where they can be combined using a data mining algorithm. The end result is an ensemble of models or a hierarchical model that combines local models built from data at each of the distributed sites. Another approach is to move data from a number of distributed sites to a central location. Once the data is centralized, a single model or ensemble of models is

built using a data mining algorithm. Some systems also support a hybrid strategy in which both the data and the models might be moved.

Data grids are grid-based infrastructures<sup>3</sup> for working with data. A grid is a distributed collection of computing resources that appears as a single virtual computing infrastructure by sharing security services, including single log-on, an LDAP-based information infrastructure, and resource management services. A data grid adds data-specific services.<sup>4</sup> These include grid FTP for moving data over a grid and data replication services, so that data might be efficiently cached over a grid.

The Semantic Web ([www.w3.org/2001/sw](http://www.w3.org/2001/sw)) extends the Web's HTML infrastructure to include semantic information defined by XML and the Resource Description Framework (RDF). The Semantic Web also interoperates with protocols such as SOAP, which is a serialization protocol so that Semantic Web client applications can access remote objects over the Web. RDF views information as a directed labeled graph and serializes it in XML (see [www.w3.org/1999/04/WebData](http://www.w3.org/1999/04/WebData)). Less formally, RDF codes information using subject-verb-object triples. For example, [www.ncar.ucar.edu/ccm/1/1](http://www.ncar.ucar.edu/ccm/1/1), Temperature, 45.5 is a subject-verb-object triple giving the temperature for a particular data record specified by a URL. The Semantic Web also supports ontologies so that we can use data taxonomies.

### Reference

1. R.L. Grossman, M. Hornick, and G. Meyer, "Emerging KDD Standards," *Comm. ACM*, Special Issue on Data Mining, Aug. 2002.
2. H. Kargupta and P. Chan, eds., *Advances in Distributed and Parallel Knowledge Discovery*, AAAI Press/MIT Press, Menlo Park, Calif., 2000.
3. I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, San Francisco, 1999.
4. A. Chervenak et al., "Protocols and Services for Distributed Data-Intensive Science," *Proc. Advanced Computing and Analysis Techniques in Physics*, Am. Inst. Physics, College Park, Md., 2000, pp. 161-163.

## The DataSpace structure

Distributed data mining systems are agent-based systems that either move data or statistical models between sites as one stage in the process of building a statistical model on distributed data. The Semantic Web extends the Web's HTML infrastructure to include semantic information defined by XML and the Resource Description Framework (RDF) and include services to work with this semantic information. Data webs provide just some of the functionality of Semantic Web. Data grids, unlike data webs, work with data using a virtual distributed computer's infrastructure. For this reason, the assumption with data grids is that you have logged on and are au-

thorized to examine and compute with remote data (see the "Data Webs and Grids" sidebar).

The viewpoint proposed here is that a data web supporting the ability to view, explore, and merge remote and distributed data is sufficient for the data mining process's initial phases and, therefore, that data webs have the potential for fundamentally changing the way scientists and engineers work with other peoples' data.

One way to make sense of the different technologies is to place them along two axes (see Table 1). Along the horizontal axis is what you do with the data (an action), such as viewing it, mining it, or computing with it. Along the vertical axis is the object of the action, which might

**Table 1. Technologies for exploring remote and distributed numerical data.**

	View	Mine and discover	Compute
Knowledge	Digital libraries	Knowledge mining	Semantic webs
Records and Attributes	Web-accessible databases	Data webs	Data grids
Files	Persistent archives	Distributed data mining	Grids

be a file, rows and columns or records and fields, or higher-order concepts such as the ontologies and related concepts underlying a knowledge management system. Data webs, data grids, and the Semantic Web can all provide Web-based access to numerical data. Data webs provide direct access to distributed rows and columns of data. Data grids enable large-scale resource sharing of computational and data resources. Semantic Webs provide knowledge-based access to data using ontologies, RDF, and agent-based architectures.

DataSpace provides a foundation for remote data analysis and distributed data mining by using four key concepts:

- Distributed columns of numerical data. DataSpace's data model is simple. Data is divided into rows (data records) and columns (data fields or attributes). Both can be distributed over the Web. DataSpace is based on a data transport protocol called the DataSpace Transport Protocol. Although data can be stored physically as files in distributed DSTP servers on a data web, data itself in DataSpace is logically just a distributed collection of columns.
- Universal correlation keys. UCKs are a globally unique ID used for relating columns of data on two different DataSpace servers. Each data column is associated with at least one column of UCKs.
- Multidimensional UCKs. UCKs can be combined to provide multidimensional keys. This is essential for working with scientific and engineering data. For example, an atmospheric data set might use a latitude UCK, a longitude UCK, and a temporal UCK to specify each sea surface temperature measurement.
- Column-based metadata. Associated with each data column is *attribute metadata* and with each data set (a collection of columns) *data set metadata*. DataSpace applications might or might not use this metadata, which is essential for building and deploying statistical models (steps 3 and 4 from earlier). DataSpace provides a simple mechanism for associating metadata to columns and collections of columns.

UCKs form the basic glue among attributes available on different DataSpace servers. All data published by a DataSpace server must be attached to at least one UCK. Every UCK is characterized by its name (not necessarily unique) and its ID number (globally unique). Suppose one DataSpace server lists the earth's surface temperature values according to longitude–latitude and another server lists precipitation values. A scientist might want to correlate precipitation with temperature. In this case the UCKs on both servers would be “latitude” (say, the ID number is 11110) and “longitude” (say, the ID number is 11111). Identical ID numbers guarantee that the key on both servers is the same and that the unit of the key on all servers is the same. For example, the ID number 11111 might specify that the key represents longitude and that longitude is measured in 1° intervals. If longitude is measured in 5° intervals, the ID number must change, although the UCK name might be “longitude” in both cases.

More precisely, UCKs allow distributed columns to be correlated in the following fashion: Pairs  $(k_i, x_i)$ , where  $k_i$  is a UCK value and  $x_i$  is an attribute value, can be combined with pairs  $(k_j, y_j)$  from another DataSpace Server to produce a table  $(x_k, y_k)$  in a DataSpace client. The DataSpace client can then, for example, find a function  $y = f(x)$  relating  $x$  and  $y$ .

DSTP uses XML to describe the metadata. For efficiency, data itself is transmitted as records delimited by carriage returns with fields delimited by commas. The open source DataSpace servers and clients we have developed communicate with DSTP. Depending on the request, DSTP servers might return one or more columns, one or more rows, or entire tables. DSTP is broadly based on the Network News Transfer Protocol, which retrieves protocol news articles (see [www.w3.org/Protocols/rfc977/rfc977.html](http://www.w3.org/Protocols/rfc977/rfc977.html)). DSTP servers use stream connections and NNTP-like commands and responses designed to accept connections from DSTP clients and to provide a simple interface to the data columns on the DSTP server. A DSTP server functions as an interface between DSTP applications and remote data.

Here is a list of the basic DSTP commands:

- METADATA [EXPAND || (CATEGORY [CategoryName] & | UCK [UCKName] & | & | SERVER [ServerName] & | DATAFILE DataFileName)]
- SET (CATEGORY CategoryName) || (UCK UCKName) || (DATAFILE DataFileName)
- SET LINE [StartLine EndLine]
- SET TYPE [ ASCII || BINARY || SOCKET || P SOCKET || SABUL ]
- SET SAMPLE RANDOM [ PERCENTAGE || LINE ]
- SET SAMPLE DECIMATE PERCENTAGE
- DATA [[Attribute ID]+ [where (Attribute ID constraint [op]\*)+ ]
- RESET [ ALL || (CATEGORY CategoryName) || (UCK UCKName/ALL) || (DATAFILE DataFileName || LINE || TYPE ASCII/BIN(ARY) || SAMPLE ]
- STOP
- QUIT
- HELP [ Command | STATUSCODE | ERRORCODE ]

The DSTP commands let DSTP clients retrieve metadata; specify UCKs, data sets, ranges, and sampling parameters; and retrieve the specified data. The metadata includes information about the units and how the data was collected and processed, the minimum and maximum values for each attribute, and other simple statistical information about the attributes.

The hard part about working with data over the Web is deciding what data to transport, what units the data is in, what processes were used to prepare the data, and how to normalize the data so it can be used with other data. These are all essential for exploratory data analysis and data mining. DSTP servers support services that return data set and attribute metadata to answer these questions. Each DSTP server also has a special file, the catalog file, containing metadata about the data sets on the server to facilitate searching and locating remote data sets.

DSTP clients and servers support

- UCK-based queries. DSTP client and server operations are based on UCKs. For example, a DSTP client can request all UCKs from a DSTP server, set a UCK, and then request all columns associated with that UCK.
- Range-based queries. Ranges might be determined using a single UCK or using several UCKs.

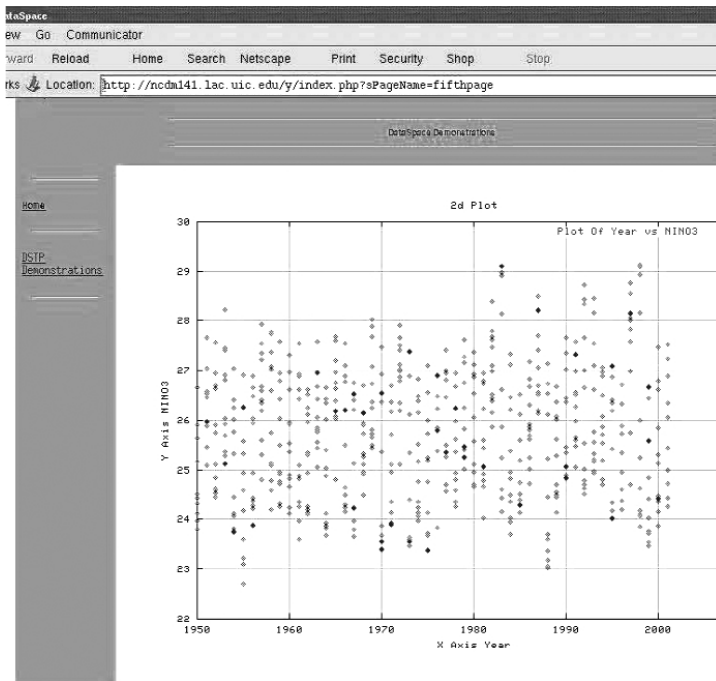
- Metadata queries. DSTP servers automatically create metadata about the data they serve and provide a simple mechanism for user-supplied metadata. DSTP clients can request attribute-based metadata, data set-based metadata, or metadata summarizing all the datasets managed by the DSTP server. For example, the metadata associated with an attribute contains the number of rows and the minimum and maximum values. This simplifies the work of DSTP client applications, such as those supporting EDA.
- Server-side sampling. DSTP servers can easily overwhelm DSTP client applications with data. The DSTP servers support server-side sampling so that the appropriate amounts of data are returned.
- Merging of distributed columns. A key benefit of DataSpace is the ability to work with distributed columns. DSTP servers pair data columns with their UCKs, and DSTP client applications can merge distributed columns by their UCKs.
- Support for missing values. DSTP servers and clients support missing values as a primitive data type. This is important for exploratory data analysis and data mining applications. If improperly handled, missing values can easily give misleading results.

*The challenge is to make these corrections as easy as pointing and clicking.*

### A simple example

A recent article in *Science* noted a relation between El Niño and the outbreak of cholera.<sup>2</sup> Although finding many text articles on the Web about El Niño and cholera is easy, finding either topic in a format in which we could readily check this hypothesis was extremely difficult. Some cholera data is available from the World Health Organization (WHO), but not the data used in the study. Although El Niño data is available, proper data is not easy to locate, and when you do locate it, it's only available in HTML or via FTP, neither of which can be directly correlated. The challenge is to make these correlations as easy as pointing and clicking—the same criterion we expect today when viewing remote documents.

To explore data mining from this perspective, we imported atmospheric data from the National Center for Atmospheric Research (NCAR) and



**Figure 1. Result of a DataSpace query on El Niño data. Using universal correlation keys and the DataSpace client's basic EDA capabilities, we can compare the El Niño data to cholera data with a few points and clicks, even though the data is from different sites and originally in quite different formats.**

cholera data from the WHO into DataSpace.

Importing this data is relatively simple. DataSpace clients and servers communicate using DSTP. In this example, an open source DSTP server managed the data; and the client was a simple Web browser. Putting data into DataSpace consists of three steps:

1. The owner can place the data in simple ASCII files with attributes delimited by separators such as “|” and records delimited by carriage returns. These data files are placed in the DSTP server data directory. This is analogous to placing html documents in an HTTP server's doc directory. DSTP servers can also access data from databases using Open Data Base Connectivity, Java Database Connectivity, or from native file formats, such as netcdf.
2. The owner can identify certain columns of data as UCKs. In practice, each application domain has its own conventions for data, which naturally lead to UCKs. The NCAR data uses a common NCAR format—a  $1^\circ \times$

$1^\circ$  latitude–longitude grid at one month intervals from 1870 to 1998. For this example, there are three UCKs for each attribute.

3. The owner places an XML file containing the attribute and file metadata in the same directory.

Querying data in DataSpace is equally simple. Here is what happens between the DSTP client and server in a typical query:

- The user opens two DataSpace sites: one containing sea surface temperature data and the other containing El Niño data.
- The DSTP server tells the DSTP client what UCKs are present and displays them, or in this case, latitude, longitude, and time.
- The user selects UCKs of interest.
- The DSTP client shows what data columns (fields) are available for these UCKs—in this case, sea surface temperature and El Niño anomaly data. The DSTP client and server accomplish this by exchanging the relevant XML metadata (showing what data column (fields) are available).
- The user selects columns of interest from these (distributed) data columns.
- The DSTP client and server exchange metadata about the file size and the client processing and visualization capabilities. The DSTP server provides server-side sampling as required for the client. Both the client and server use the Predictive Model Markup Language (PMML) conventions for working with missing values (see [www.dmg.org](http://www.dmg.org)).
- The DSTP server streams the relevant columns (and associated UCKs) to the DSTP client, which merges the streams to create the desired data set. The default is for DSTP servers to not store the data in XML. In many cases, this dramatically reduces the server's storage, computation, and bandwidth requirements.
- The client performs simple EDA of the selected columns, displays simple visualizations, and builds simple models (see Figure 1).

The interaction between the DSTP client and server proceeds through three phases: UCKs are retrieved, then metadata, and then data. By using UCKs, we can compare different (possibly distributed) columns. By next retrieving the metadata, the client and server can select an appropriate amount of data to return in the final step. The DSTP servers we have developed treat the UCKs, metadata, and data quite differently. Dif-

ferent storage formats and caching policies can be used by the DSTP servers for each. For example, the UCKs and metadata can be stored in XML and kept in memory during operation, whereas the data files can be stored more efficiently and retrieved on a per query basis.

The client handles the basic transformations directly and automatically. For known UCKs, we can create simple direct mechanisms by hand to implement the transformations required using, for example, the templated transformations that are part of PMML Version 2.0. Although this might be counterintuitive, interesting distributed data queries can be done with this approach. The reason is that a basic set of UCKs can sometimes cover a large class of data in a community. Think of this as the 80–20 rule for UCKs. The alternative of using ontologies and RDF, although much more flexible and powerful, creates a higher barrier before data can be put by the researcher or user into a data web (see [www.w3.org/2001/sw](http://www.w3.org/2001/sw)). As an analogy, you could argue that requiring the Web to use URLs, RDF, and ontologies (which are equally applicable to documents and make searches easier) would have significantly lowered the likelihood of the Web's success.

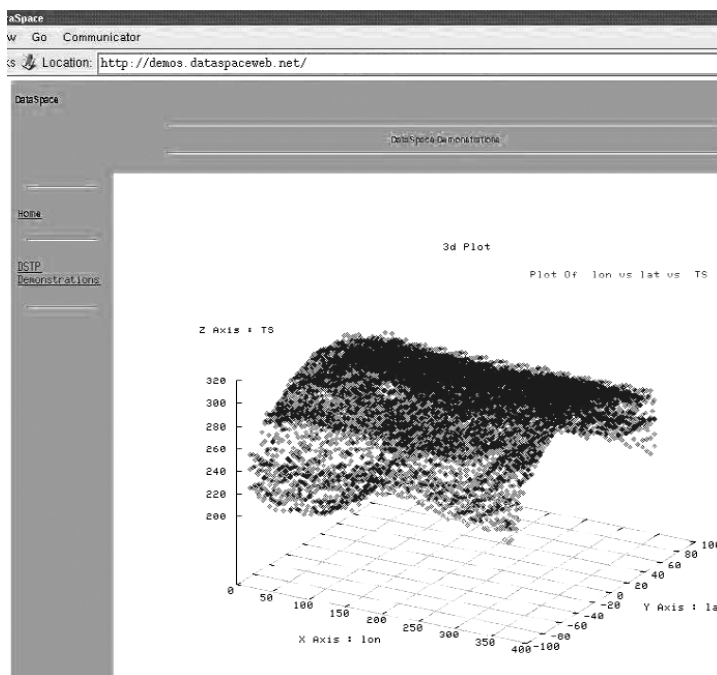
### DataSpace examples

We have developed a number of prototype applications that use the DataSpace infrastructure. Here we describe three such applications that demonstrate the adaptability of the infrastructure with various large data sets.

#### Earth science data

We placed approximately 100 Gbytes of Community Climate Model (CCM3) data from the NCAR on a DSTP server. Scientists use CCM3 data to study CO<sub>2</sub> warming and climate change, climate prediction and predictability, atmospheric chemistry, paleoclimate, biosphere-atmosphere transfer, and nuclear winter scenarios. The data contains monthly satellite measurements of global surface temperatures, precipitation, ozone levels, and a vegetation index.

There are three UCKs for this application: latitude, longitude, and time. Figure 2 shows the result of a DataSpace query for sea surface temperature. The DSTP client for this application supports queries by UCK, attribute range, attribute, and data set. The client can view the data, download the data, graph the data, and do simple EDA operations on the data.



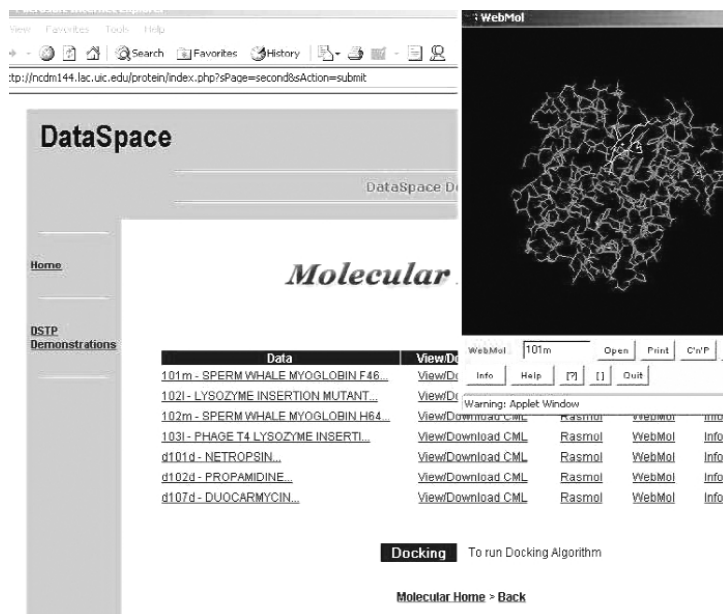
**Figure 2. Result of a DataSpace query for sea surface temperature on National Center for Atmospheric Research data from a DSTP server.**

The DSTP client can also compare the CCM3 data and overlay it with other data sets sharing one or more of the same UCKs.

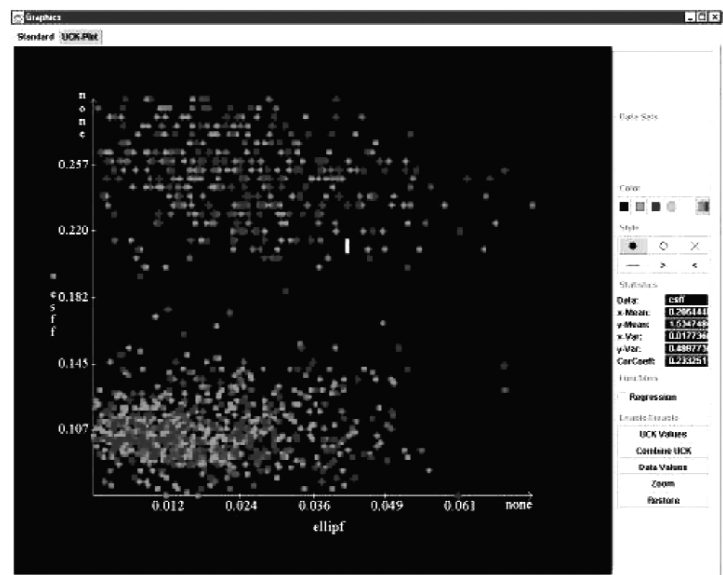
#### Protein data

In this application,<sup>3</sup> we took the data from the Protein Data Bank (PDB) and placed it in a DSTP server in Halifax. The data consisted of records of the form C,PRO,2,28.901,38.374,3.596. In this example, C (Carbon) is the type of atom, PRO (Proline) is the residue to which the atom belongs, 2 is the ID of the amino acid molecule, and the final three coordinates are the *x*, *y*, and *z* coordinates of the atoms serving as the UCKs. We also placed data describing drugs on a DSTP server in Amsterdam using the same UCKs. Both these DSTP servers were on a testbed for DataSpace that let us measure the performance of various queries involving large data sets.

The DSTP client application in Figure 3 can retrieve and interactively explore proteins. The proteins can be displayed in the Chemical Markup Language or in PDB file format. The data can also be visualized using a graphics program such as Rasmol or a Web visualization tool such as Webmol.



**Figure 3.** Result of a distributed DataSpace query. One of the sites contains 3D protein data from the Protein Data Bank, which we replicated in Halifax to test the DataSpace infrastructure. The other contains 3D data describing small organic compounds, which might be used as drugs. This data is on a DataSpace server in Amsterdam, which is also part of our testbed. The query results in the docking of a candidate drug molecule with a protein.



**Figure 4.** The result of DataSpace query of astronomical data from two sky surveys. One data set is the Two Micron All Sky Survey data from a DSTP Server at CalTech and the other is the Digital Palomar Observatory Sky Survey data, which we replicated on a DSTP Server in Chicago.

Of more interest, we can do a distributed query between a protein molecule from the Halifax DSTP server and a small organic compound from the Amsterdam DSTP server. For example, Figure 3 shows a drug candidate molecule docking with a protein from the PDB. In addition, the DSTP client can query the National Center for BioTechnology Information's PubMed for all references related to the proteins and drugs on the DSTP clients.

### Astronomical data

In this example taken from Robert Grossman and his colleagues' study,<sup>4</sup> we queried two geographically distributed astronomical source catalogs: 2MASS (Two Micron All Sky Survey) data from a DSTP server at the California Institute of Technology and DPOSS (Digital Palomar Observatory Sky Survey) data, which we replicated on a DSTP server in Chicago. The 2MASS data is in the optical wavelength (0.4–0.7 micron) range, while the DPOSS data is in the infrared (1.2–2.2 micron) range. The light source's position in both data sets is in polar coordinates. The UCKs are the right ascension and declination, measured in degrees.

A typical query of interest to astronomers is of the form, "Find all pairs, one from the DPOSS catalog and one from 2MASS, whose angular separation is less than a given tolerance." Figure 4 illustrates a DataSpace query with a tolerance of two seconds in the region of the sky with right ascension from 183° to 270° and declination from 17° to 47° (somewhere over the North Pole).

After working with the examples previously described and related examples for the past two years, we decided to standardize on certain primitives for normalizing, transforming, and aggregating data. The following four transformations seem sufficient to cover most transformations we use in practice: normalization, discretization, value mapping, and aggregation. We have been working with the Data Mining Group to standardize these operations, all four of which are part of the PMML Version 2.0. Today, DataSpace clients perform these transformations in an ad hoc fashion. The next version of DataSpace will support DMG-compliant implementations of these operations.

Paraphrasing DMG's description, the approach to transformations is not to cover the full set of preprocessing functions that we might need to collect and prepare the data for mining, but rather to introduce a templated set of basic operations designed to cover most common operations (see [www.dmg.org](http://www.dmg.org)). The operations cover, for example, the normalization of input values required for neural networks and the types of quantile range discretizations used to transform skewed data. Indeed, it has been our experience that they already cover many transformations we use in practice to prepare data for mining. Correlating El Niño anomalies with cholera outbreaks requires aggregating and normalizing the data from the two data sets so we can compare the data meaningfully. Today, our DSTP clients essentially hard-code some of the standard transformations for common UCKs. We feel that together with the basic graphics and visualization that are already part of DataSpace, these transformations will cover many desired EDA operations, further enhancing the ability of scientists using DataSpace to casually explore remote and distributed data.

Think of this work as a specific Semantic Web application and implementation. We faced the challenge of understanding what specific services, data, and metadata are required for exploratory data analysis using data webs and implementing these services in a scalable manner. Any metadata required for data mining can be put into XML. Based on our experiences building data web applications, we have been an active participant in the DMG, and our data web applications use the XML metadata standards developed by them.

We have chosen not to use the Semantic Web's RDF standard because our interest is in data mining, not knowledge management. Although in some sense, data is a trivial type of knowledge, from a practical viewpoint, having a scalable robust web infrastructure to work with data is useful—for the same reason we still have databases and data archive systems even though knowledge management and AI systems should have made these “trivial” long ago.

Webs are built from services—DSTP is a Web service for working with remote and distributed data. Recently, we released a Web Service Definition Language for DSTP. This will be useful as semantic web applications become more common. We chose to implement DSTP directly instead of over the Simple Object Access Protocol for several reasons: SOAP was not

around when we started and doesn't scale as well to high-volume data streams. We will soon release a version of DSTP that uses SOAP and that is suitable for DSTP client applications involving small data sets. ❧

## References

1. *Proc. 7th ACM SIGKDD Int'l. Conf. Knowledge Discovery and Data Mining*, 2001, ACM Press, New York, 2001.
2. P. Pascual et al., “Cholera Dynamics and El Niño-Southern Oscillation,” *Science*, vol. 289, Sept. 2000, pp. 1766–1769.
3. D. Hamelberg and R.L. Grossman, *A DataSpace Infrastructure for Bioinformatics Data*, tech. report, Laboratory For Advanced Computing, Univ. of Illinois, Chicago, 2000.
4. R. Grossman et al., “A DataSpace Infrastructure for Astronomical Data,” *Data Mining for Scientific and Eng. App.*, Kluwer Academic Publishers, New York, 2001.

**Robert Grossman** is a professor of mathematics, statistics and computer science at the University of Illinois at Chicago and the director of the National Center for Data Mining at UIC. He is also president of the Two Cultures Group. His research interests are data mining, data-intensive computing, data webs, high performance computing, high-performance networking, and scientific computing. He has an AB in mathematics from Harvard and a PhD in applied mathematics from Princeton. He is a member of the IEEE, ACM, ACS, and SIAM. Contact him at the Univ. of Illinois at Chicago, 322 SEO, MC 249, 851 S. Morgan St., Chicago, IL 60607; [grossman@uic.edu](mailto:grossman@uic.edu).

**Marco Mazzucco** is a postdoctoral research associate at the National Center for Data Mining at the University of Illinois at Chicago. His research interests include data mining, high-performance data management, high-performance computing, scientific computing, machine learning, and high-performance networking. He has a BS in mechanical engineering and a PhD in mathematics from the UIC. Contact him at the Univ. of Illinois at Chicago, 322 SEO, MC 249, 851 S. Morgan St., Chicago, IL 60607; [marco@lac.uic.edu](mailto:marco@lac.uic.edu).

For more information on this or any other computing topic, please visit our digital library at <http://computer.org/publications/dlib>.