cells will become a reasonable strategy as the principal regulatory mechanisms of endothelial haemodynamic dysfunction that are associated with the disease process become better understood. Because the haemodynamics and location of lesions are defined spatially, transcriptional profiling of endothelial cells as a function of location is a promising approach to this goal. With some refinement of methods of cell isolation, it should be possible to profile endothelial cells at sites of lesion susceptibility in intact arteries.

Acknowledgments

The authors' work is supported by NIH MERIT Award HL36049 (PFD), NIH grant HL62250 (PFD) and a Whitaker Foundation Biomedical Engineering Research Grant (NDP).

References

1 Zarins, C. K., Giddens, D. P., Bharadvaj, B. K., Sottiurai, V. S., Mabon, R. F. and Glagov, S. (1983) *Circ. Res.* 53, 502–514
2 Davies, P. F. (1995) *Physiol. Rev.* 75, 519–560
3 Resnick, N. and Gimbrone, M. A., Jr (1995) *FASEB J.* 9, 874–882
4 Davies, P. F. *et al.* (1997) *Annu. Rev. Physiol.* 59, 527–549
5 Resnick, N., Collins, T., Atkinson, W., Bonthron, D. T., Dewey, C. F. and Gimbrone, M. A., Jr (1993) *Proc. Natl. Acad. Sci. U. S. A.* 90, 4591–4595
6 Shyy, J. Y., Lin, M. C., Han, J., Lu, Y., Petrime, M. and Chien, S. (1995) *Proc. Natl. Acad. Sci. U. S. A.* 92, 8069–8073
7 Davies, P. F., Remuzzi, A., Gordon, E. S., Dewey, C. F. and Gimbrone, M. A., Jr (1986) *Proc. Natl. Acad. Sci. U. S. A.* 83, 2114–2118
8 Topper, J. N., Cai, J., Falb, D. and Gimbrone, M. A., Jr (1996) *Proc. Natl. Acad. Sci. U. S. A.* 93, 10417–10422
9 Topper, J. N., Wasserman, S. M., Anderson, K. R., Cai, J., Falb, D. and Gimbrone, M. A., Jr (1997) *J. Clin. Invest.* 99, 2942–2949
10 Topper, J. N. *et al.* (1997) *Proc. Natl. Acad. Sci. U. S. A.* 94, 9314–9319
11 DePaola, N., Gimbrone, M. A., Davies, P. F. and Dewey, C. F., Jr (1992) *Arterioscler. Thromb.* 12, 1254–1257
12 DePaola, N., Davies, P. F., Pritchard, W. F., Florez, L., Harbeck, N. and Polacek, D. C. (1999) *Proc. Natl. Acad. Sci. U. S. A.* 96, 3154–3159
13 Nagel, T., Dewey, C. F. and Gimbrone, M. A., Jr *Arterioscler. Thromb. Vasc. Biol.* (in press)
14 Liang, P. and Pardee, A. B. (1992) *Science* 257, 967–971
15 Barbee, K. A., Mundel, T., Lal, R. and Davies, P. F. (1995) *Am. J. Physiol.* 268, H1765–H1772
16 Davies, P. F., Mundel, T. and Barbee, K. A. (1995) *J. Biomech.* 28, 1553–1560
17 Walpola, P. L., Gotlieb, A. I., Cybulsky, M. I. and Langille, B. L. (1995) *Arterioscler. Thromb. Vasc. Biol.* 15, 2–10
18 Nakashima, Y., Raines, E. W., Plump, A. S., Breslow, J. L. and Ross, R. (1998) *Arterioscler. Thromb. Vasc. Biol.* 18, 842–851
19 Nagel, T., Resnick, N., Atkinson, W. J., Dewey, C. F. and Gimbrone, M. A. (1994) *J. Clin. Invest.* 94, 885–891
20 Geiger, R. V., Berk, B. C., Alexander, R. W. and Nerem, R. M. (1992) *Am. J. Physiol.* 262, C1411–C1417
21 Shen, J., Luscinskas, F. W., Connolly, A., Dewey, C. F. and Gimbrone, M. A., Jr (1992) *Am. J. Physiol.* 262, C384–C390
22 Falcone, J. C., Kuo, L. and Meininger, G. A. (1993) *Am. J. Physiol.* 264, H653–H659
23 Ranjan, V. and Diamond, S. L. (1993) *Biochem. Biophys. Res. Commun.* 196, 79–84
24 Martin-Mondiere, C. F., Caprani, A., Desgranges, P. C., Loisance, D. Y. and Charron, D. J. (1989) *ASAIO Trans.* 35, 288–290
25 Eberwine, J. *et al.* (1992) *Proc. Natl. Acad. Sci. U. S. A.* 89, 3010–3014
26 Eberwine, J. *et al.* (1995) *Prog. Brain Res.* 105, 117–126
27 Schena, M., Shalon, D., Davis, R. W. and Brown, P. O. (1995) *Science* 270, 467–470
28 DeRisi, J. *et al.* (1996) *Nat. Genet.* 14, 457–460
29 Madison, R. and Robinson, G. (1998) *BioTechniques* 25, 504–514

Kleisli: a new tool for data integration in biology

Su Yun Chung and Limsoon Wong

One of the central problems in bioinformatics is data retrieval and integration. The existing biological databases are geographically distributed across the Internet, complex and heterogeneous in data types and data structures, and constantly changing. With the current rapid growth of biomedical data, the challenge is how large volumes of data retrieved from multiple databases can be transformed and integrated automatically and flexibly. This article describes a powerful new tool, the Kleisli system, for complex queries across multiple databases and data integration.

In the era of high-throughput technologies such as whole-genome sequencing, gene-expression profiling and combinatorial drug discovery, large volumes of biomedical data are routinely generated, and this demands rapid data processing. The challenge to modern bioinformatics is how to automate the processes of information retrieval and integration.

Biological data are inherently complex, ranging from plain-text nucleic acid and protein sequences, through the three-dimensional structures of therapeutic drugs and macromolecules, and high-resolution images of cells and tissues, to microarray-chip outputs. The data in various autonomous databases are organized in extremely heterogeneous formats, ranging from flat-file format (plain text files, binary files) through relational data models to highly nested data model such as ASN-1 [adopted by the US National Center for Biotechnology Information (NCBI)]. Moreover, the

S. Y. Chung (suchung@sdsc.edu) is at the Department of Biochemistry and Molecular Biology, Uniformed Services University of the Health Sciences, 4301 Jones Bridge Road, Bethesda, MD 20814-4799, USA and is currently at KRIS Technology, STE #2, 713 Santa Cruz Ave, Menlo Park, CA 04025, USA. L. Wong (limsoon @krdl.org.sg) is at Kent Ridge Digital Labs, 21 Heng Mui Keng Terrace, Singapore 119613.

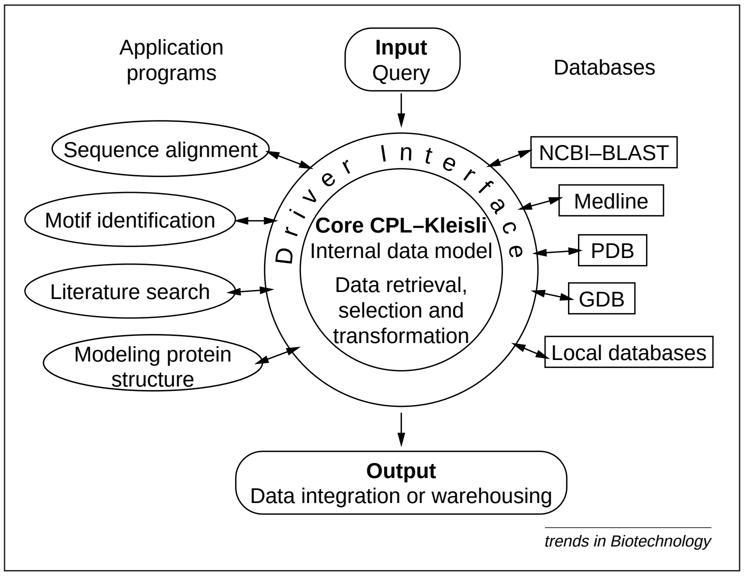


Figure 1
A schematic diagram of the Kleisli system and data integration. The core of the Kleisli system is collection programming language (CPL) and the Kleisli query engine, which performs the tasks of parsing, optimizing and executing queries. The Kleisli query engine talks to the outside data sources or application programs via driver interfaces. For a typical query, the drivers perform the tasks of connecting requests to the remote Internet database servers or to locally installed data sources or application programs. The drivers then retrieve the output data and transform them into the CPL data-exchange format. The retrieved data from multiple heterogeneous sources are evaluated, transformed and integrated by the CPL–Kleisli core.

data structures are constantly evolving to reflect new research and technology development.

The heterogeneous and dynamic nature of these biological databases present major obstacles to the use of data mining to investigate specific biological problems. Clearly, simple retrieval of data is not sufficient for data mining; it requires flexible data manipulation and sophisticated data integration. The tasks involved include: complex queries across multiple heterogeneous data sources; data warehousing by merging data derived from multiple public sources and local (private) sources; and multiple data-analysis procedures that require feeding subsets of data derived from different sources into various application programs for gene finding, protein-structure prediction, functional-domain or motif identification, phylogenetic-tree construction, graphic presentation and so on.

The ability to pass selected data rapidly and smoothly between multiple and sequential steps is essential. Recently, several review articles[1–3] have discussed the challenges for bioinformatics and observed that many existing biological-data-retrieval systems are not fully up to the demand of flexible and painless data integration[4,5]. These articles have briefly mentioned the Kleisli system as a tool for complex queries across public databases. However, we wish to provide a fuller account of the current state of the Kleisli system and to introduce the strengths and features of the system to the biology community.

Kleisli

The Kleisli system (http://www.kris-inc.com/; http://adenine.krdl.org.sg:8080/publications/) was developed as a general solution to broad-scale data-integration problems[6,7]. The Kleisli system was implemented on top of modern functional programming technology[8] and the fundamental aspects of the Kleisli architecture have been described previously[7,9]. In brief, the core of the Kleisli system is a high-level query language called collection programming language (CPL)[9,10], which supports a powerful data model and makes data transformation, manipulation and integration easy (Fig. 1).

A query is typically written as a CPL program, which is then parsed, optimized and executed by the Kleisli engine. The Kleisli system 'talks' to the outside data sources and application programs via the driver interface (Fig. 1). Drivers are program scripts that perform the task of connecting to specific data sources or application programs, sending queries in the language of corresponding database-management systems (DBMSs) and transforming the retrieved data into the internal CPL data model. The Kleisli–CPL system is able to model data in diverse formats including plain text files, the popular relational or object-oriented data models, application programs and web-based data sources.

Data integration in the Kleisli system is based on the underlying query language, which, in addition to a rich data model, offers many high-level operations such as collection comprehensions[9,10] and pattern matching. This greatly facilitates the task of selecting, manipulating and combining specific data sets retrieved from diverse data sources for analysis. Thus, Kleisli–CPL supports complex queries and data retrieval across distributed heterogeneous data sources and functions as an efficient data transformer and integrator.

Today, there are more than a hundred major biological databases and application servers on the Internet[11,12] and new sites are being introduced at an ever-increasing rate. The data structures of these databases are heterogeneous and constantly evolving. In order to explore the wealth of information available on the Web, we need tools that can efficiently retrieve and integrate information across networks of data sources. A system that merely provides an interface to a collection of databases and analysis software will not be useful in many cases if it requires tedious programming to make use of the interface, as is the case with CORBA[13]. Moreover, very few of the existing multidatabase query systems address the problem of data integration after data retrieval.

Kleisli goes one step further: it can perform complex queries, data transformations and data integration on these biological data sources. The operations can be carried out in an extremely straightforward manner that does not require extensive programming. Many complicated bioinformatics problems that require using multiple databases and analysis programs have a simple expression in CPL and are efficiently executed by Kleisli.

A problem: integrate BLAST search results with the annotated features of a GenPept report

Instead of generalizing, we feel that the most direct way to appreciate the power and flexibility of the Kleisli system is to see how it works in a real example. The most common practice for rapidly identifying putative functions of a new protein sequence is to conduct a database search using the BLAST program[14] or another similarity-search program in a large database such as the nonredundant protein-sequence database (NR) curated at NCBI (http://ncbi.nlm.nih.gov/).

Typically, BLAST sends back the search results to the user in a plain text file that describes a hit list of subject sequences that match the query sequence with a score above the user-specified cut-off value, along with sequence alignments for the matching regions between the query sequence and the subject sequences[15]. The output of BLAST can be made more informative if it also provides some annotated features about the matching regions; for example, we might like to know whether the matching regions contain specific protein functional domains or active sites.

One way to accomplish this task is by manually navigating through the Web links. First, send a query sequence for BLAST search and go through the sequence alignments in the BLAST report. For each matching pair, the BLAST report provides a hypertext-linked accession number or uid (unique identification) for the GenPept report. The GenPept reports from the Entrez database are also presented to the user as plain text[5]. Users have to look up the annotated features to find out whether any of the annotated protein regions or sites also fall into the corresponding matching region in the BLAST report. This procedure can be carried out manually if no more than a few sequences are involved but, when the number of sequences increases to hundreds or higher, this becomes a daunting, if not impossible, task. Fortunately, this simple selection and integration of data derived from two data sources, the BLAST and Entrez databases, can be accomplished automatically by the Kleisli system.

Box 1. A CPL program that integrates a BLAST search result with the annotated features of the corresponding GenPept report

Code

```
 1 webblast-blastp-detail (#name: ‘nr-blast’, #db: ‘nr’, #level: 1);
 2 [(#uid: x.#uid, #accn: x.#accession, #title: x.#title,
 3 #query-start: qs, #query-end: qe,
 4 #feat-start: f.#start, #feat-end: f.#end,
 5 #identities: i.#Matching-Percentage, #feat-anno: f.#anno)
 6 | \x <- process SEQ using nr-blast,
 7 \t <- aa-get-seqfeat-by-uid (x.#uid),
 8 \h <- - - x.#hits, h.#pscore <= 1.0E~8,
 9 \f <- t.#feature, f.#start >= h.#subjectstart, f.#end <=
     h.#subjectend,
10 \qs == (f.#start - h.#subjectstart) + h.#querystart,
11 \qe == (f.#end - h.#subjectend) + h.#queryend];
```

Explanation

Line 1 instructs Kleisli to make a connection to the BLASTP server via the specified driver. Kleisli then processes the BLAST search using this connection and the user-input query sequence SEQ (line 6).

The BLAST report returns a list of hits according a user-specified scoring-cut-off value (1.0E~8; line 8).

Line 7 instructs Kleisli to invoke a driver (aa-get-seqfeat-by-uid) to access the Entrez database and obtain the corresponding GenPept report t for each matching subject sequence x [with a unique identifier (uid) of x.#uid] returned by the BLAST search.

Lines 8–11 involve the integration of selected data from the BLAST and the GenPept reports. Kleisli looks into the GenPept report t and finds every annotated feature region f that is strictly contained within the aligned region h in the BLAST report.

Kleisli then selects the subset of data that satisfies the constraints and transforms and integrates this retrieved data into its internal data model as a list of records containing the uid, accession, title, start and end points of the aligned region ‘h’ and feature region ‘f’, percentage sequence identity, and feature annotations (lines 2–5).

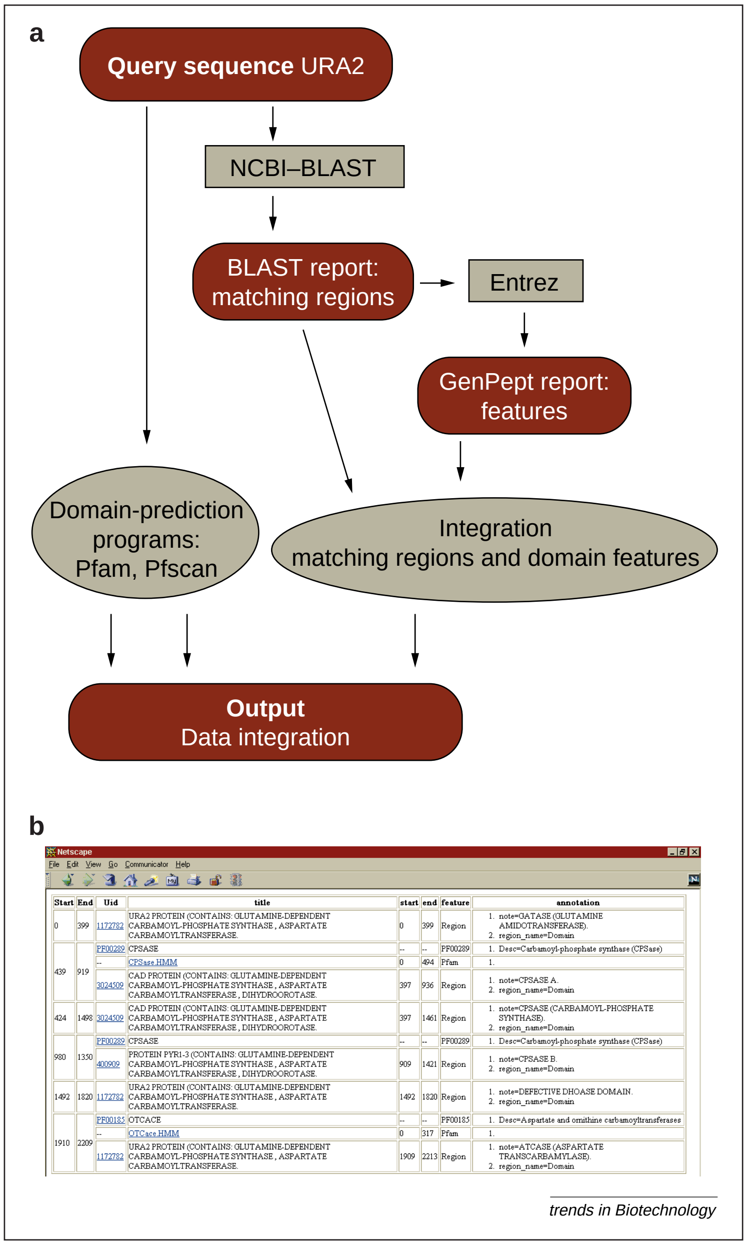
An example query

The yeast URA2 protein was chosen as an example query sequence to integrate the BLAST results and the annotated features of the GenPept report. Yeast URA2 is a multifunctional protein with four separate enzymatic activities, involved in the *de novo* pyrimidine-biosynthesis pathway[16–19]. It is 2214 amino acids long and organized into four different domains – the glutamine-aminotransferase domain, the carbamoyl-phosphate-synthase domain, the dihydro-orotase domain and the aspartate-transcarbamoylase domain, which is homologous to ornithine-transcarbamoylase domains from other proteins. These domain structures are conserved from yeast to human.

The CPL program implementing the queries and data integration across the BLAST and Entrez databases is straightforward and is shown in Box 1. The syntax may appear to be alarming at first glance but the logic is simple. The CPL program instructs Kleisli to perform a complex query across two databases (BLAST and Entrez) and provides pattern-matching and simple string-manipulation facilities to accomplish the task of data selection and integration. First, it identifies the ‘common sequence regions’ that are shared by the aligned regions of the BLAST report and the feature-annotation regions of the GenPept report, and then selects and returns the subset of data that satisfy the constraints.

The returned data are then transformed and integrated into the Kleisli–CPL internal data model as a list of records containing information on the matching regions of the BLAST results and their corresponding annotated features derived from the GenPept report. For further analysis, we also feed the query sequence URA2 into two other domain-prediction application programs (Pfam[20] and Pfscan[21]) using Kleisli drivers interfacing to these programs. The combined results of domain structures described by the three independent methods (the BLAST–Entrez data-integration approach as implemented by the CPL program in Box 1, the Pfam analysis and the Pfscan analysis) are shown in Fig. 2, with a flow diagram illustrating the entire data-integration process; Fig. 2b shows a small portion of the output from Kleisli (http://sdmc.krdl.org.sg/kleisli/psZ/cdw-featureblast.ps).

This simple example demonstrates how Kleisli accesses, manipulates and integrates information derived from multiple data sources and application programs. We want to emphasize that the entire task was accomplished by a dozen lines of CPL and is executed automatically and rapidly without any human intervention. In addition to developing queries on protein sequences, a similar strategy can be applied to DNA sequences, and more complicated data-query and -integration tasks can be carried out with similar ease and flexibility. The implementation of other queries would be similar to this example in their complexity and program length. There is no real limit to the number of databases or analysis programs that can be used.


a
Query sequence URA2
NCBI–BLAST
BLAST report: matching regions
Entrez
GenPept report: features
Domain-prediction programs: Pfam, Pfscan
Integration matching regions and domain features
Output Data integration
b
Netscape
File Edit View Go Communicator Help

Start	End	Uid	title	start	end	feature	annotation
0	399	1172782	URA2 PROTEIN (CONTAINS: GLUTAMINE-DEPENDENT CARBAMOYL-PHOSPHATE SYNTHASE , ASPARTATE CARBAMOYLTRANSFERASE.	0	399	Region	1. note=GATASE (GLUTAMINE AMIDOTRANSFERASE). 2. region_name=Domain
439	919	PF00289	CPSASE	--	--	PF00289	1. Desc=Carbamoyl-phosphate synthase (CPSase)
		--	CPSase.HMM	0	494	Pfam	1.
		3024509	CAD PROTEIN (CONTAINS: GLUTAMINE-DEPENDENT CARBAMOYL-PHOSPHATE SYNTHASE , ASPARTATE CARBAMOYLTRANSFERASE , DIHYDROOROTASE.	397	936	Region	1. note=CPSASE A. 2. region_name=Domain
424	1498	3024509	CAD PROTEIN (CONTAINS: GLUTAMINE-DEPENDENT CARBAMOYL-PHOSPHATE SYNTHASE , ASPARTATE CARBAMOYLTRANSFERASE , DIHYDROOROTASE.	397	1461	Region	1. note=CPSASE (CARBAMOYL-PHOSPHATE SYNTHASE). 2. region_name=Domain
980	1350	PF00289	CPSASE	--	--	PF00289	1. Desc=Carbamoyl-phosphate synthase (CPSase)
		400909	PROTEIN PYR1-3 (CONTAINS: GLUTAMINE-DEPENDENT CARBAMOYL-PHOSPHATE SYNTHASE , ASPARTATE CARBAMOYLTRANSFERASE , DIHYDROOROTASE.	909	1421	Region	1. note=CPSASE B. 2. region_name=Domain
1492	1820	1172782	URA2 PROTEIN (CONTAINS: GLUTAMINE-DEPENDENT CARBAMOYL-PHOSPHATE SYNTHASE , ASPARTATE CARBAMOYLTRANSFERASE.	1492	1820	Region	1. note=DEFECTIVE DHOASE DOMAIN. 2. region_name=Domain
1910	2209	PF00185	OTCACE	--	--	PF00185	1. Desc=Aspartate and ornithine carbamoyltransferases
		--	OTCace.HMM	0	317	Pfam	1.
		1172782	URA2 PROTEIN (CONTAINS: GLUTAMINE-DEPENDENT CARBAMOYL-PHOSPHATE SYNTHASE , ASPARTATE CARBAMOYLTRANSFERASE.	1909	2213	Region	1. note=ATCASE (ASPARTATE TRANSCARBAMYLASE). 2. region_name=Domain

trends in Biotechnology

Figure 2

(**a**) A flow diagram illustrating the data-integration process. (**b**) A section of a direct screen dump of the Kleisli results for the yeast URA2 protein. This shows the domains predicted by the combination of integrating the data from a BLAST search and GenPept feature annotations, and the Pfam and Pfscan domain-prediction programs.

Kleisli can pass data smoothly and flexibly between multiple sequential steps. The Kleisli system is robust and efficient in its handling of parallel multiple-database queries owing to its build-in query optimizer.

What are the options?

Several other approaches have been available to explore information in heterogeneous biological databases and to address the problems of multiple-database query and data integration.

Hypertext links

One of the simplest and most successful approaches to integrating biological databases is by connecting heterogeneous databases via hypertext links on the World Wide Web (WWW) and providing comprehensive indexing systems for query[3]. Such a hypertext-navigation approach is adopted by the sequence-retrieval system SRS[4] and by DBGET–LinkDB[22]. These systems are convenient to use for simple operations but offer no facilities for the flexible transformation and integration of data derived from heterogeneous sources. Indeed navigating around the hypertext links is like flipping from one entry in an encyclopedia to another, with only one flip of the pages at a time[23]. Complex queries that require selection and transformation of structural data across large numbers of heterogeneous databases are almost prohibitive for this type of hypertext navigation approach. It requires a significant amount of manual work for data integration.

TAMBIS

The TAMBIS (transparent access to multiple bioinformatics information sources)[24] project aims to provide transparent access to multiple databases and analysis tools using a knowledge-driven graphic user interface (GUI) for query formulation. The knowledge-driven GUI[24] is implemented on top of an old version of the Kleisli system for the actual execution of database queries and data exchange. The TAMBIS query system is user friendly and provides a simple way for non-programmers to specify queries, but the kind of query that can be expressed is severely restricted by the designs of the TAMBIS query templates.

*OPM*QS*

The OPM*QS (object–protocol model multidatabase query system)[25,26] uses the common data-model approach to accomplish the task of multiple-database query and data integration. OPM*QS provides GUIs, a structured query language, an object data model and sophisticated data-management tools. However, it lacks a simple data-exchange model and relies on a dictionary to map individual database schema into the common OPM global schema. As a result, OPM*QS requires more effort and planning to add new sources than Kleisli does.

IGD

The most demanding approach is that followed by the integrated genome database (IGD)[27]. IGD essentially adopts the data-warehousing approach to integrate the 20 or so major data sources in the domain of genome projects and molecular biology. It has a global schema that is very different from those of the underlying data sources. For each data source to be integrated, IGD has to convert the source data format into its global data model and store the converted data in a centralized local database. IGD provides a global schema, a popular GUI and the ACEDB[28] data-management and -query facilities.

However, it has several potential problems. First, the need to store the integrated data locally limits the number of data sources that can be integrated. As the number of integrated data sources increase, it is likely to push ACEDB beyond its design limits of size and performance. Second, the cost of maintaining the system is high; it is extremely difficult to adjust the global schema when new data sources are added or old data sources are removed or evolved.

Kleisli

By contrast, the Kleisli system takes a distributive-database approach to data integration and does not require the maintenance of a locally integrated data warehouse, as is the case for IGD. It is indifferent to database schema and does not require a global schema for data transformation, as OPM★QS does.

The strength of the Kleisli system resides in its powerful internal data model. Kleisli has a self-describing data-exchange format to convert various data types into its data model. Its nested, relational data model encompasses the conventional relational data model as well as other data models frequently used in bioinformatics. Therefore, biological data in diverse data formats can be readily converted into the Kleisli internal data model. Furthermore, Kleisli is equipped with an advanced type-inference system that can deduce the structure of input and output data directly from the structure of a query. The powerful data model, the data-exchange format and the type-inference system remove the need for global schema in Kleisli, making Kleisli the most flexible tool available for complex queries across multiple heterogeneous databases and data integration. The limitation is that the Kleisli system does require users to understand the CPL query language. To simplify usage, we are experimenting with a GUI for routine queries[29] and a more robust GUI is now under development.

Conclusion

We have demonstrated how the Kleisli system implements complex bioinformatic problems in a simple way and significantly reduces the effort and difficulty involved in data integration. The Kleisli system is an ideal tool for data-format exchange, project-oriented data integration and data warehousing, which uses large volumes of data derived from multiple public sources and local private sources. Moreover, the Kleisli system is particularly suited to data-driven research and discovery processes; users are likely to have unpredictable needs, such as rapid access to absolutely current data, ad-hoc access to new data sources and the use of a variety of computational tools for data analysis. The current Kleisli system is robust and has more than 50 drivers interfacing to major biological databases and application programs. For data integration in bioinformatics, it offers versatility, flexibility, clarity, control and performance.

Acknowledgments

We would like to thank J. Chen, T. Kandasamy, K. Lin, A. Ting and J. Wang for implementing the queries, and H. Liu, A. Kolatkar and S. Subbiah for comments. This work was supported by a grant from the US National Science Foundation (INT9725507) and the Bioinformatics Center at the National University of Singapore and Kent Ridge Digital Labs. KRIS Technology now markets the Kleisli system as BioKRIS. The opinions or assertions contained herein are the private ones of the authors and are not to be construed as official or reflecting the views of the US Department of Defense or the Uniformed Services University of the US Health Services.

References

1 Baker, P. G. and Brass, A. (1998) *Curr. Opin. Biotechnol.* 9, 54–58

2 Benton, D. (1996) *Trends Biotechnol.* 14, 261–272

3 Karp, P. D. (1996) *Trends Biotechnol.* 14, 273–279

4 Etzold, T., Ulyanov, A. and Argos, P. (1996) *Methods Enzymol.* 266, 114–128

5 Schuler, G. D., Epstein, J. A., Ohkawa, H. and Kans, J. A. (1996) *Methods Enzymol.* 266, 141–162

6 Buneman, P., Davidson, S., Hart, K., Overton, C. and Wong, L. (1995) *Proceedings of the 21st International Conference on Very Large Data Bases*, pp. 58–169, Morgan–Kaufman

7 Davidson, S., Overton, C., Tannen, V. and Wong, L. (1997) *Int. J. Digit. Libr.* 1, 36–53

8 Appel, A. W. (1992) *Compiling with Continuations*, Cambridge University Press

9 Buneman, P., Naqvi, S., Tannen, V. and Wong, L. (1995) *Theoret. Comput. Sci.* 149, 3–48

10 Wadler, P. (1992) *Math. Struct. Comput. Sci.* 2, 461–493

11 Ashburner, M. and Goodman, N. (1997) *Curr. Opin. Genet. Dev.* 7, 750–756

12 Karp, P. D. (1998) *Trends Biochem. Sci.* 23, 114–116

13 Siegel, J. (1997) *CORBA: Fundamentals and Programming*, Wiley

14 Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990) *J. Mol. Biol.* 215, 403–410

15 Madden, T. L., Tatusov, R. L. and Zhang, J. (1996) *Methods Enzymol.* 266, 131–141

16 Souciet, J. L., Potier, S., Hubert, J. C. and Lacroute, F. (1987) *Mol. Gen. Genet.* 207, 314–319

17 Souciet, J. L., Nagy, M., Le Gouar, M., Lacroute, F. and Potier, S. (1989) *Gene* 79, 59–70

18 Nagy, M., Gouar, M., Potier, S., Souciet, J. L. and Herve, G. (1989) *J. Biol. Chem.* 264, 8366–8374

19 Smith, T. (1998) *Trends Genet.* 14, 291–293

20 Sonnhammer, E. L., Eddy, S. R., Birney, E., Bateman, A. and Durbin, R. (1998*) Nucleic Acids Res.* 26, 320–322

21 Bucher, P., Karplus, K., Moeri, K. and Hoffman, K. (1996) *Comput. Chem.* 20, 3–24

22 Fujibuchi, W. *et al.* (1997) *Pacific Symp. Biocomput.* 3, 683–694

23 Gelbart, W. M. (1998) *Science* 282, 659–661

24 Baker, P. G., Brass, A., Bechhofer, S., Goble, C., Paton, N. and Stevens, R. (1998) *Intell. Syst. Mol. Biol.* 6, 25–34

25 Chen, I. M. A. and Markowitz, V. M. (1995) *Inform. Syst.* 20, 393–418

26 Markowitz, V. M., Chen, I. M. A. and Kosky, A. (1996) *Theoretical and Computational Genome Research* (Suhai, S., ed.), pp. 161–176, Plenum Press

27 Ritter, O., Kocab, P., Senger, D., Wolf, D. and Suhai, S. (1994) *Comput. Biomed. Res.* 27, 97–115

28 Durbin, R. and Thierry-Mieg, J. (1994) *Computational Methods in Genome Research* (Suhai, S., ed.), pp. 45–56, Plenum Press

29 Tan, W. C. and Wang, K. (1998) *J. Database Management* 9, 24–32

Pharmacogenomics poster

This issue of *Trends in Biotechnology* contains a poster detailing the field of Pharmacogenomics. Our thanks to the sponsors and to all who contributed to the poster.