



## caCORE: A common infrastructure for cancer informatics

Peter A. Covitz\*, Frank Hartel, Carl Schaefer, Sherri De Coronado, Gilberto Fragoso, Himanso Sahni, Scott Gustafson and Kenneth H. Buetow

National Cancer Institute Center for Bioinformatics, National Institutes of Health, U.S. Department of Health and Human Services, 6116 Executive Boulevard, Suite 403, Rockville MD 20852, USA

Received on December 20, 2002; revised on April 23, 2003; accepted on June 19, 2003

### ABSTRACT

**Motivation:** Sites with substantive bioinformatics operations are challenged to build data processing and delivery infrastructure that provides reliable access and enables data integration. Locally generated data must be processed and stored such that relationships to external data sources can be presented. Consistency and comparability across data sets requires annotation with controlled vocabularies and, further, metadata standards for data representation. Programmatic access to the processed data should be supported to ensure the maximum possible value is extracted. Confronted with these challenges at the National Cancer Institute Center for Bioinformatics, we decided to develop a robust infrastructure for data management and integration that supports advanced biomedical applications.

**Results:** We have developed an interconnected set of software and services called caCORE. Enterprise Vocabulary Services (EVS) provide controlled vocabulary, dictionary and thesaurus services. The Cancer Data Standards Repository (caDSR) provides a metadata registry for common data elements. Cancer Bioinformatics Infrastructure Objects (caBIO) implements an object-oriented model of the biomedical domain and provides Java, Simple Object Access Protocol and HTTP–XML application programming interfaces. caCORE has been used to develop scientific applications that bring together data from distinct genomic and clinical science sources.

**Availability:** caCORE downloads and web interfaces can be accessed from links on the caCORE web site (<http://ncicb.nci.nih.gov/core>). caBIO software is distributed under an open source license that permits unrestricted academic and commercial use. Vocabulary and metadata content in the EVS and caDSR, respectively, is similarly unrestricted, and is available through web applications and FTP downloads.

**Contact:** [covitzp@mail.nih.gov](mailto:covitzp@mail.nih.gov)

**Supplementary information:** <http://ncicb.nci.nih.gov/core/> publications contains links to the caBIO 1.0 class diagram and the caCORE 1.0 Technical Guide, which provide detailed information on the present caCORE architecture, data sources and APIs. Updated information appears on a regular basis on the caCORE web site (<http://ncicb.nci.nih.gov/core/>).

### INTRODUCTION

A critical factor in the advancement of biomedical research is the ease with which data can be integrated, redistributed and analyzed both within and across domains. A number of obstacles have challenged the bioinformatics community's ability to achieve semantic consistency across data sets and to provide standardized programmatic access to those data. The uneven application of controlled vocabulary, the lack of metadata standards, and inconsistent deployment of data services and interfaces have conspired to create what has been referred to as the bioinformatics equivalent of neighboring but isolated medieval nation states, each with different systems of weights and measures (Stein, 2002).

The mission of the National Cancer Institute Center for Bioinformatics (NCICB) is to provide informatics infrastructure and scientific applications that support advanced translational research in cancer biology and medicine. We are charged with providing consistent, open access to the multitude of diverse data sets that arise from NCI-sponsored initiatives. These data come from studies involving cell and molecular biology, genomics, histopathology, drug development, and clinical trials. We must integrate those data with each other, and with the broader body of public biomedical information.

Here we describe caCORE, the NCICB core infrastructure for biomedical informatics. In addition to providing facilities for data management and redistribution, caCORE helps solve problems of data integration. caCORE consists of a series of component technologies and services. Vocabulary

\*To whom correspondence should be addressed.

services, a metadata registry and object-oriented middleware each provide caCORE with facilities for rendering data from diverse sources interoperable and semantically consistent. We describe how a prototype application that makes use of caCORE provides new insights into biomedical problems, and discuss how caCORE can be used to support other research efforts.

## ENTERPRISE VOCABULARY SERVICES

Enterprise Vocabulary Services (EVS) forms the semantic underpinnings of caCORE. EVS organizes distinct but overlapping terminologies and provides a rich controlled vocabulary for coding and retrieval. EVS produces the NCI Thesaurus, a cancer-focused vocabulary built using the description logics paradigm, and the NCI Metathesaurus, a collection of biomedical vocabularies that is based on the National Library of Medicine (NLM) Unified Medical Language System (UMLS) (Lindberg *et al.*, 1993; McCray and Nelson, 1995; Bechhofer and Goble, 2001). The subject domain of both vocabularies is the same, encompassing various basic and clinical research areas as well as publishing and administrative functions. The content of each is different, however, because of the different data models and the different uses. The content of the NCI Metathesaurus and the NCI Thesaurus is available for unrestricted use by any organization. Up to date information on the EVS component of caCORE can be found at <http://ncicb.nci.nih.gov/core/EVS>

### NCI Metathesaurus

The UMLS is focused on medical terminologies and categorizations, but the needs of the cancer research community extend beyond medical terminology to basic biology and a wide range of translational and applied research fields. The NCI Metathesaurus was conceived to fulfill these broader needs. To create the NCI Metathesaurus, we modify the UMLS Metathesaurus by dropping vocabularies and classifications (sources), with low relevance to cancer research and treatment, and by adding sources with high relevance. The current version of the NCI Metathesaurus contains 56 sources, and is based on the UMLS release 2002AC. We update several times per year as new releases of the UMLS become available. The NCI Metathesaurus is available for direct interactive use through a web-based interface (<http://ncimeta.nci.nih.gov>), which also has a complete list of source vocabularies. Programmatic access is available via the caBIO application programming interfaces (APIs).

### NCI Thesaurus

The NCI Thesaurus is the principal vocabulary product built by the NCI. It contains approximately 25 000 concepts, represented by about 70 000 terms. The original content of the Thesaurus was collected by an exhaustive survey that brought together for the first time the classification schemes,

informal vocabulary and naming conventions used in much of the cancer research community. Formal governance processes have been instituted that assure the content of the NCI Thesaurus remains comprehensive and current.

The NCI Thesaurus is organized into 21 hierarchical trees covering areas such as Neoplasms, Drugs, Anatomy, Genes, Proteins, Techniques and administrative terminology. It is intended for use by applications needing tightly controlled vocabulary, hierarchy traversal and traversal of relationships among concepts. The NCI Thesaurus is presently available for download as a tab-delimited flat file and as XML (<ftp://ftp1.nci.nih.gov/pub/cacore/EVS>). Interactive web access is available through a browsing application that illustrates the description logic relationships embedded in the semantic structure (<http://nciterms.nci.nih.gov>).

### Vocabulary development and server environment

The contents of the NCI Metathesaurus and the NCI Thesaurus are developed and served using products from Apelon, Inc. We use these commercial tools because, for the present, there are no open source equivalents that fully support a multi-editor description logic authoring environment. However, the NCICB is actively supporting the development of such open alternatives by academic groups. In the meantime we are committed to wrapping the commercial terminology server functionality into publicly accessible open source APIs in caBIO.

## CANCER DATA STANDARDS REPOSITORY

Common Data Elements (CDEs) are metadata that define data elements used in research studies. The NCI launched the CDE initiative to address the lack of data standards in cancer clinical trials (Silva *et al.*, 2001). The project has expanded to include a variety of cancer research domains, including cancer prevention trials, biomarker studies, biomedical imaging and genomics.

As the effort to define CDEs expanded, we adopted a standard for metadata organization and structure, the ISO/IEC 11179 standard for metadata registry implementation (<http://metadata-stds.org>). ISO/IEC 11179 defines a number of Administered Components and associated attributes. The central Administered Component is a *Data Element*, which is constructed from a *Data Element Concept* (semantic component) and a *Value Domain* (representational component). The *Value Domain* has an associated set of *Permissible Values* that constrain the type and content of a given datum that can be collected in an actual research study. Administered Components each belong to a *Context*, which in our implementation is used to separate them into sub-domains of cancer research that have a common metadata curatorial process and authority.

It is important to note the relationship and distinction between controlled vocabulary and standardized metadata, as many in the bioinformatics community seem to confuse or equate the two. Only the fully specified CDE with

resolvable semantic and representational components is a unit of metadata adequate for use in creating interoperable systems. Standard vocabularies and ontologies, however, have an essential role to play in the creation of CDEs: CDE names, definitions and *Permissible Values* are derived from terminology found in EVS.

The Cancer Data Standards Repository (caDSR) is a database and application tool set that implements the ISO/IEC 11179 standard, and is used as a central registry for the CDEs (<http://ncicb.nci.nih.gov/core/caDSR>). The database and application technology is based upon a Data Element Registry product from Oracle Corp. and uses Oracle 8i and Oracle 9iAS. The software is available upon request. Nonetheless, we encourage outside institutions to use the NCI-hosted caDSR tools rather than use the caDSR code base to create their own registry. Our goal is for the caDSR instance hosted by the NCI to become a singular life sciences metadata registry, open for use by all interested parties. Any group wishing to curate a novel set of metadata can either join an existing Context or ask to have a new Context created for them. We believe this arrangement will minimize problems with CDE redundancy and divergence, and will promote widespread re-use of CDEs.

The CDE content of the caDSR is available for unrestricted use by any organization. Access to the CDEs is available through the interactive CDE Browser (<http://ncicb.nci.nih.gov/cdebrowse>). The CDE Browser supports searching and bulk export of CDEs in spreadsheet or XML format. A documented PL/SQL API to the system is also available ([ftp://ftp1.nci.nih.gov/pub/cacore/caDSR/caCORE1.0\\_caDSR\\_API.pdf](ftp://ftp1.nci.nih.gov/pub/cacore/caDSR/caCORE1.0_caDSR_API.pdf)). A new API based on open source Cancer Bioinformatics Infrastructure objects (caBIO) technology is planned for the next major caCORE release.

## CANCER BIOINFORMATICS INFRASTRUCTURE OBJECTS

The caBIO model and architecture is at the heart of caCORE. caBIO was initially intended to meet the need for programmatic access to the information at several NCI projects, including the Cancer Genome Anatomy Project (CGAP; <http://cgap.nci.nih.gov>), the Cancer Molecular Analysis Project (CMAP; <http://cmap.nci.nih.gov>), the Genetic Annotation Initiative (GAI; <http://gai.nci.nih.gov>), clinical trials databases, and other publicly available repositories (Riggins and Strausberg, 2001; Schaefer *et al.*, 2001; Strausberg *et al.*, 2001; Buetow *et al.*, 2002). Its success has led to its adoption as the primary architecture for data federation and integration at the NCI. We believe that caBIO could be extended to be a general infrastructure for biomedical informatics.

### Modeling and programming techniques

Design of the caBIO domain object model is driven by use cases. We represent the model using the Unified Modeling

Language (UML). Use case development and modeling is performed using Rational Rose (Rational Software, Inc.) and elements of the Rational Unified Process and Extreme Programming methodologies (Beck, 1999; Kruchten, 2000). The caBIO 1.0 domain object UML class diagram is available on the Supplementary Information site for this article (<http://ncicb.nci.nih.gov/core/publications>).

We use an automatic code generator to generate Java source code from UML (<http://www.saic.com/quava>). Modifications to the object model are always made at the UML-level, and not directly in the source code. If a new underlying database schema is needed, this can also be automatically generated from the UML. If an existing schema must be supported, then manual coding is needed to implement the connections between the object and data layers. The resulting Java software consists of packages that are named with the prefix *gov.nih.nci.caBIO*. A Java archive (JAR) file, source code and example client programs are downloadable under an open-source license (<http://ncicb.nci.nih.gov/core/caBIO>). The software is instantiated in an *n*-tier architectural design (Fig. 1).

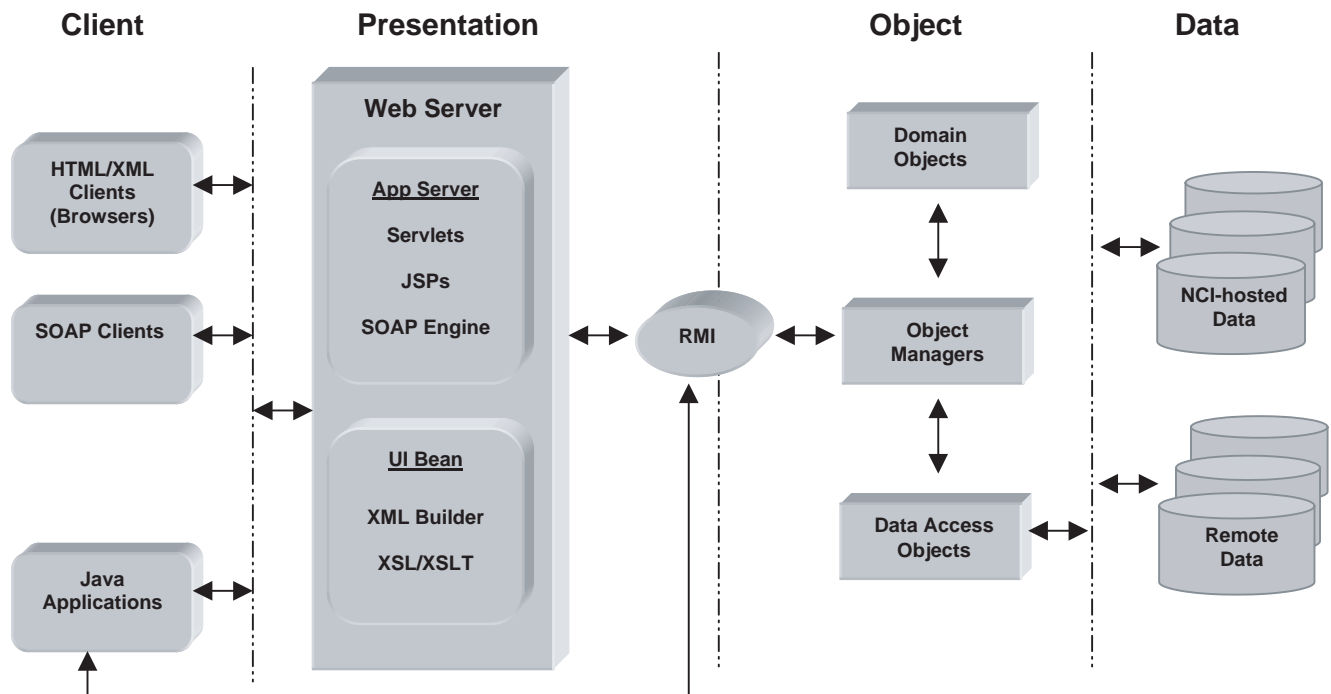
### Data layer

At the foundation of the architecture is a data layer that includes all caCORE data sources (Table 1). Both data warehouse and data federation strategies are employed, and flat file, relational and internet protocol sources are supported. caBIO contributes a data warehouse that is used to integrate several of the sources listed in Table 1. The content of the caBIO data warehouse is refreshed approximately bi-weekly. The caBIO data warehouse is presently implemented in Oracle 8i, but does not make use of any Oracle-specific functions, and thus could presumably be instantiated on other SQL-based platforms. EVS contributes its vocabulary server database while the caDSR supplies its CDE database, each incorporated using a data federation strategy. The human genome database is federated from a remote source, the Distributed Annotation System (DAS) server hosted by the University of California, Santa Cruz (Dowell *et al.*, 2001; Kent *et al.*, 2002).

### Object layer

Domain objects represent biological, laboratory and clinical entities (Table 2). These objects have attributes that are part of the objects themselves as well as relationships to other objects as indicated by the caBIO class diagram (<http://ncicb.nci.nih.gov/publications>). We are exploring mechanisms for describing these attributes and relationships as standard metadata that can be stored in the caDSR.

All domain objects implement the *java.io.Serializable* interface and *gov.nih.nci.caBIO.util.XMLInterface*, thus facilitating their transport to the Presentation Layer as XML. Three additional interfaces are also defined: *Expressable*,



**Fig. 1.** The caCORE *n*-tier system architecture as instantiated at the NCI. The Data layer consists of files, relational databases and other resources that reside on NCI or outside hosts. The Object layer contains biomedical domain objects that support the public APIs, data access objects for retrieval from the Data layer, and object managers to regulate the traffic. Access to the Object layer goes through Java RMI. The Presentation layer consists of a web server, application server, SOAP engine and XML processing beans. API access is available for Java clients via RMI, and for all clients via the Presentation layer.

**Table 1.** caCORE human and mouse data sources

Data type	Source (institution)
Genes, transcripts, sequences and homologs	Unigene (NCBI); LocusLink (NCBI); Homologene (NCBI)
Expression (EST, SAGE, microarray)	Cancer Genome Anatomy Project (NCI); NCI60 project (NCI, Stanford); Director's Challenge Initiative (NCI)
Polymorphisms	Genetic Annotation Initiative (NCI)
Sequence trace files	Genetic Annotation Initiative (Washington University, Incyte, Agencourt)
Clones and libraries	Cancer Genome Anatomy Project (NCI); dbEST (NCBI); IMAGE Consortium
Molecular anomalies in cancer	Cancer Molecular Analysis Project (NCI)
Cytogenetic map locations	Unigene (NCBI)
Human Genome	Golden Path via DAS (UCSC)
Pathways	BioCarta Pathways (BioCarta)
Vocabularies, ontologies and functional classification	NCI Metathesaurus (NCI); NCI Thesaurus (NCI); CMAP Ontology (NCI); Gene Ontology (Gene Ontology Consortium)
Common Data Elements	Cancer Data Standards Repository (NCI)
Therapeutic agents and targets	Cancer Therapy Evaluation Program (NCI); Developmental Therapeutics Program (NCI); Division of Cancer Prevention (NCI)
Clinical trial protocols	Cancer Therapy Evaluation Program (NCI); Special Programs of Research Excellence (NCI); Division of Cancer Prevention (NCI)

*Ontologable* and *Relationable*. Currently, only the *Gene* class implements *Expressable*. This interface defines a single method, *getExpression()*, that returns an array of *Expression-Experiments* containing expression levels for the Gene.

The *Organ*, *Disease*, *CMAPOntology* and *GoOntology* classes all implement the *Ontologable* interface, as each of these object types defines entities occurring in ontologies or taxonomies such as the Gene Ontology or other

Table 2. caBIO domain objects

Object	Definition
Agent	Therapeutic drug, intervention or therapy
Anomaly	Irregularity in either the expression or sequence of a Gene in diseased versus normal tissue sources
Chromosome	Chromosome from a particular species Taxon
ClinicalTrialProtocol	Protocol associated with a clinical trial. Provides trial information such as organization ID, participants, phase and administered Agent
Clone	I.M.A.G.E. clone object; provides access to sequence, associated trace files and the clone's source library
CMAPOntology	CMAP gene ontology, which categorizes genes by function; provides access to Gene objects corresponding to the ontological term, as well as to ancestor and descendant terms in the ontology tree
CMAPOntologyRelationship	Child or parent relationship between CMAPOntology objects
ConceptSearch	Searchable concept term in a controlled vocabulary occurring in the NCI Metathesaurus; used to find synonym or semantic types for the concept of interest
ConsensusSequence	A specialization of Sequence; represents the consensus of a set of Contigs
Contig	One of the set of overlapping sequence fragments used to assemble a ConsensusSequence
Disease	Disease objects specify a disease name and ID
DiseaseRelationship	An object specifying a child or parent relationship between Disease objects
ESTExperiment	Data from an ExpressionExperiment using expressed sequence tags. Derived from ExpressionExperiment
ExpressionExperiment	An abstract class; parent of ESTExperiment and SAGEExperiment
ExpressionFeature	Associated with a Gene object through the Gene's getExpressionFeature() method; provides access to the list of Organs where the gene is known to be expressed
ExpressionMeasurement	Absolute or relative amount of a given molecule
ExpressionMeasurementArray	Array of ExpressionMeasurement objects
Gene	Based on a Unigene cluster, this central object in the model has relationships to many other domain objects
GeneAlias	An alternative name for a Gene; provides descriptive information about the gene (as it is known by this alias), as well as access to the Gene object it refers to
GeneHomolog	Defined only in relation to another Gene object of interest. Derived from Gene class, and also provides the percent of sequence similarity of the homolog to the related Gene object
GoOntology	Gene object's position in the Gene Ontology, as recorded by LocusLink; provides access to Gene objects annotated with the GO term, as well as to ancestor and descendant terms in the ontology tree
GoOntologyRelationship	Child or parent relationship between GoOntology objects
Histopathology	Anatomical changes in a diseased tissue sample; captures the relationship between organ and disease
Library	cDNA library. In our instantiation, data is primarily from CGAP and other cancer-related libraries in dbEST
MapLocation	Physical map location of a Gene
Organ	Organ whose name occurs in a controlled vocabulary; provides access to any Histopathology objects for the organ, and, because it is ontologable, provides access to its ancestral and descendant terms in the vocabulary
OrganRelationship	Child or parent relationship between Organ objects
Pathway	Cellular or biochemical pathway. In our instantiation pathways are derived from BioCarta, and are associated with Gene that comprise the pathways
Protein	Protein; provides access to the encoding gene via its GenBank Accession, the taxon in which this instance of the protein occurs, and references to homologous proteins in other species
ProteinHomolog	Defined only in relation to another Protein object of interest. ProteinHomolog is derived from the Protein class and also provides the percent of sequence similarity of the homolog to the related protein of interest. B21
Protocol	Laboratory protocol used to prepare a Library
ProtocolAssociation	An association class relating ClinicalTrialProtocols to Diseases
ReadSequence	The output of a TraceFile object, an ASCII representation of the nucleotide sequence
SAGEExperiment	Serial analysis of gene expression (SAGE) data. Derived from ExpressionExperiment
Sequence	Nucleotide sequence; provides access to the clones from which it was derived, the ASCII representation of the residues it contains, and the sequence ID
SNP	Single Nucleotide Polymorphism; provides access to the clones and the trace files from which it was identified, the two most common substitutions at that position, the offset of the SNP in the parent sequence, and a confidence score
Target	A Gene thought to be at the root of a disease etiology, and which might be a target for therapeutic intervention in a clinical trial
Taxon	Species name. Can be scientific or common name
Tissue	A group of similar cells united to perform a specific function
TraceFile	Sequence trace file

sources (Ashburner *et al.*, 2000). Each of the domain objects implementing the *Ontologable* interface implements the *getChildRelationships()* and *getParentRelationships()* methods, which can be applied to get the parent and descendant terms in the vocabulary where the object's name is defined. The return type of both of these methods is actually an array of *Relationable* objects, objects that implement the *Relationable* interface. The *Relationable* interface provides methods to set, search and retrieve relationships between objects. Classes implementing the *Relationable* interface include *OrganRelationship*, *DiseaseRelationship*, *CMAPOntologyRelationship* and *GoOntologyRelationship*. Each instance of a relationship object stores its relationship *type* (child/parent) and the set of *Ontologable* objects participating in that relationship.

Access to the object layer occurs via Java Remote Method Invocation (RMI), but network and persistence details are abstracted away from developers that use the caBIO APIs. Object managers broker requests for data from domain objects to the data access objects, which in turn retrieve the data from the data layer and pass it back to the presentation layer or to the client directly, depending on the source of the request. We are exploring the possibility of using alternative transport protocols in the future in order to work around firewall issues that can make RMI problematic for some client sites.

### Presentation layer

The presentation layer is built upon a web server infrastructure that includes Apache HTTP Server (<http://httpd.apache.org>), Tomcat (<http://jakarta.apache.org/tomcat/index.html>), Zope (<http://www.zope.org>) and Apache SOAP (<http://ws.apache.org/soap>). Java servlets and server pages deliver content to client applications. All caBIO objects can be transformed into serialized XML representations. We use eXtensible Stylesheet Language (XSL) and XSL Transformations (XSLT) to transform object data for use in various displays and to support the SOAP API. XML linking language (Xlink) is used to control the amount of information returned from a single XML call.

## APPLICATION PROGRAMMING INTERFACES

caBIO provides the primary public API to caCORE. EVS and caDSR do have their own interfaces, but we are wrapping those into caBIO in order to provide a uniform, consistent programmatic access. Three APIs to caBIO are available, each suitable for different client programming environments. Detailed documentation of the APIs, beyond what is described below, can be found in the caCORE Technical Guide (<http://ncicb.nci.nih.gov/core/publications>).

### Java API

caCORE offers direct access to the domain objects through the Java API. Developers can download the caBIO.jar file and use its packages directly in their programs. When a caBIO

object is instantiated in a local client program, it is transparently populated with data via RMI calls to the caBIO server. Typically a client will want to instantiate particular objects that satisfy some criteria. Each class in the *gov.nci.nih.caBIO.bean* package has a companion *SearchCriteria* class that provides methods for search and retrieval. For example, the *Gene* and *Pathway* classes have corresponding *GeneSearchCriteria* and *PathwaySearchCriteria* classes, respectively. In order to issue a query, one instantiates and sets parameters in a *SearchCriteria* object that corresponds to the domain object of interest.

*SearchCriteria* query parameters can be set to include attributes of the domain object of immediate interest as well as those of related objects. As an example, consider how one would look for biological pathways that include the BRCA1 gene. First, the 'symbol' parameter of a *GeneSearchCriteria* object is set to 'BRCA1'. Then, the *GeneSearchCriteria* object is set as a parameter for a *PathwaySearchCriteria* object. When the pathway query is issued, pathways that contain the BRCA1 gene are returned in the form of a list of *Pathway* objects. The SOAP API also supports this ability to query by the attributes of a related domain object (see below and Fig. 2).

Accessing vocabulary concepts in the NCI Metathesaurus using the caBIO Java API follows the same search paradigm deployed by caBIO for other domain objects. The *ConceptSearch* and *ConceptSearchCriteria* classes provide the needed functionality for vocabulary queries. Instances of both classes are first created, and a term of interest is set as an attribute of the *ConceptSearchCriteria* object using its *setSearchTerm* method. The *search* method of the *ConceptSearch* object is then invoked with the *ConceptSearchCriteria* object as a parameter; this operation returns an array of *Concept* objects matching the search criteria. To query the retrieved vocabulary *Concept* objects, the caBIO API currently provides *get* methods for concept name, synonyms, semantic types and vocabulary sources. API support for the NCI Thesaurus and other individual vocabularies, which reside in a distinct server environment from the NCI Metathesaurus, is planned for the next major caCORE release.

### SOAP web services

The SOAP web services standard is an XML-based protocol that enables remote procedure calling and data exchange between applications written in any programming language that has a SOAP client module. Using the SOAP client packages, developers can format caBIO requests as an XML document that is posted to the caBIO SOAP server. The server receives the request and forwards the request to the appropriate object in the *gov.nih.nci.caBIO.webservices* package. The caBIO server processes the requests and returns the results of the query as an XML document embedded in the SOAP envelope. Figure 2 illustrates how to use the SOAP::Lite module in Perl to retrieve biological pathways

```

#!/usr/bin/perl

## Demonstration of a caBIO SOAP request from Perl.  Retrieves Pathways
## that contain the BRCA1 gene from the caBIO server.

use SOAP::Lite;
use HTML::Entities;

# Set server, port
$server = "cabio.nci.nih.gov";
$port = "80";

# Initialize service
$URI='urn:nci-pathway-service';
$PROXY_PATH='/soap/servlet/rpcrouter';
$soap = SOAP::Lite
    -> uri($URI)
    -> proxy("http://$server:$port$PROXY_PATH");

# Set up $geneSearch to retrieve BRCA1 gene
my %geneSearchCriteria=();
$geneSearchCriteria{"symbol"}="BRCA1";
$geneSearch = SOAP::Data->type(map => \%geneSearchCriteria);

# Use $geneSearch as criteria for getPathways
my %pathwaySearchCriteria=();
$pathwaySearchCriteria{"GeneSearchCriteria"} = $geneSearch;
$pathwaySearch = $soap->getPathways(SOAP::Data->type(map =>
\%pathwaySearchCriteria));

# If errors, print fault code; else, print decoded result
if ($pathwaySearch->fault) {
    print $pathwaySearch->faultcode.$pathwaySearch->faultstring;
}
else {
    $xmlDoc = $pathwaySearch->result;
    print decode_entities($xmlDoc);
}

```

**Fig. 2.** Perl SOAP client that retrieves Pathways containing the BRCA1 gene. The SOAP service is initialized by creating a new SOAP::Lite object and setting the URI and proxy address. Search criteria are stored in hashes where the keys are the criteria types and the values are the query parameters. \$geneSearch contains a mapping of the \$geneSearchCriteria hash to the corresponding SOAP data type. \$geneSearch is then stored as a query parameter in the \$pathwaySearchCriteria hash. The query is sent by calling the *getPathways()* method from the *nci-pathway-service*. If no faults are encountered, the payload of the SOAP message returned from the server is decoded and printed.

that contain the BRCA1 gene from the *nci-pathway-service*. The complete list of SOAP services hosted by NCI can be found at <http://cabio.nci.nih.gov/soap/services/index.html>.

### HTTP-XML interface

Using the HTTP-XML interface, users can submit caBIO queries directly in the URL of a browser and retrieve the results of their queries in XML. The HTTP interface leverages the *GetXML* service which allows researchers to perform an operation and pass in search criteria associated

with the operation as parameters embedded in the URL. An XML document that contains the results of the query is returned. For example, a researcher can obtain an XML document containing information on the *PTEN* gene via the URL <http://cabio.nci.nih.gov/servlet/GetXML?operation=Gene&Symbol=PTEN>, where 'operation' indicates which domain object is requested, 'Symbol' is the search criterion, and 'PTEN' is the search criterion parameter value. One or more search criterion-value pairs can be included. The HTTP API provides additional features including a

mechanism to throttle the amount of data or objects returned in the XML document and the ability to format the XML results using a customized XSL style sheet. These features allow researchers to create custom views of caBIO results.

## caCORE-BASED APPLICATIONS

caCORE forms the foundation for a number of scientific and clinical applications. One of these is CMAP, a work in progress that can be regarded as a prototypical caCORE-powered application (Buetow *et al.*, 2002). The availability of caCORE enabled CMAP to be prototyped in a relatively short period of time. Cancer data and data relationships are presented in CMAP with rich graphics, and the application leverages caBIO APIs to provide a straightforward interface to quite complex underlying queries. The data are annotated in part with vocabulary found in EVS, and the clinical trials that are displayed are themselves increasingly being conducted using CDEs from the caDSR.

A typical CMAP usage scenario might be to ask the following:

1. In brain tissue from patients diagnosed with the glioblastoma multiforme subtype of astrocytoma, which genes in the p53 signaling pathway are over or under expressed in cancerous versus normal tissue? Answer: BCL2; TIMP3; GADD45A; CCND1.
2. What is the underlying evidence for such aberrant expression of, for example, CCND1? Answer: SAGE tags for cyclin D1 appear with 3-fold greater frequency in cancerous brain tissue versus normal brain tissue.
3. What gene products of the p53 signaling pathway are known targets for therapeutic agents? Answer: TP53; RB1; BCL2; CDK4; MDM2; CCNE1.
4. Are any of the agents known to target these genes being tested specifically in glioblastoma patients? Answer: No such trials listed.

After investigating the molecular profiles of the p53 signaling pathway components in greater detail, a clinical scientist might wish to consider whether agents known to target this pathway might be good candidates for use in a new trial in glioblastoma multiforme patients. The reader is encouraged to try out CMAP and other caCORE-powered scientific applications, including the Cancer Models Database and the Gene Expression Data Portal (<http://cmap.nci.nih.gov>; <http://cancermodels.nci.nih.gov>; <http://gedp.nci.nih.gov>).

## DISCUSSION

Bioinformatics data management providers must serve both the interactive web user and the bioinformatics programmer in order to be effective. Most sites provide interactive web interfaces, but support for outside programmers varies widely. FTP access to bulk data downloads is sometimes

offered, but in many cases programmatic access to the data hosted at the original site would be far more efficient and preferable. Notable pioneers on this quest to build a global network of programmatically accessible bioinformatics services include the UCSC and Ensembl groups using DAS to share the latest releases of the human genome (Dowell *et al.*, 2001; Kent *et al.*, 2002); the XEMBL project offering both HTTP-XML and SOAP-XML services for EMBL nucleotide records (Wang *et al.*, 2002); and the DNA Databank of Japan (DDBJ), offering a number of SOAP services, including Blast and Clustal as well as database record retrieval (<http://xml.nig.ac.jp/soapp.html>). Many projects, including caCORE, benefit by consuming these services rather than unnecessarily replicating them locally.

Standardization of the transmission protocols and data exchange formats is necessary but not sufficient for true system interoperability. Data content must be defined according to a common semantic understanding, and should be tagged with standard metadata descriptors. The CDEs hosted in the caDSR are developed from terms found in the EVS. This practice establishes a linkage between the semantic intent and the database representation of research data collected and coded using CDEs. To our knowledge the caDSR is the only ISO/IEC 11179 metadata registry dedicated to life sciences research subject matter. There are however a number of efforts to standardize health information management and exchange (Forrey *et al.*, 1996; Dolin *et al.*, 2001). We see the caDSR as complementary to these efforts, and hope the caDSR will become a general resource for metadata development and registration in biomedical research.

There are a number of public entry points to caCORE resources. Programmatic access is available by downloading the caBIO package, which includes the Java classes needed to use the Java API, and examples in Perl that use the SOAP API (<http://ncicb.nci.nih.gov/core/caBIO>). Web-based utilities provide direct access to the EVS and caDSR resources (<http://ncimeta.nci.nih.gov>; <http://nciterms.nci.nih.gov>; <http://ncicb.nci.nih.gov/cdebrowse>). Scientific applications that have been built upon caCORE provide immediate access to NCI-hosted data and analysis facilities (<http://cmap.nci.nih.gov>; <http://emice.nci.nih.gov>; <http://cancermodels.nci.nih.gov>; <http://gedp.nci.nih.gov>).

One can use caBIO software to serve up local data if the data types are the same as those found in the caBIO object model. The process would be to instantiate the caBIO data warehouse, load it with local data, and then connect the caBIO middleware. Scripts for this process are available upon request, and will become a standard part of the caBIO software download. If an existing local database with a different schema needs to be connected, a skilled Java programmer can modify the source code for the caBIO data access classes to work with the local data sources. In the future, we hope to provide an even more straightforward mechanism for connecting local

data sources to the caBIO data layer, including the addition of new data types.

We have observed that an increasing number of scientists without programming skills are grappling with data sets in spreadsheets that are too large to analyze through traditional interactive web forms. To address this growing constituency, we are building a novel web application that will enable columns of data in spreadsheet files to be used as search criteria in queries against caBIO, with the results returned in spreadsheet file format as well.

While caCORE was born out of the needs of the cancer research community, it is intended as a general resource. Cancer research has historically contributed to many areas beyond tumor biology. Given that data comparability and accessibility are general problems in life science, we hope that caCORE will contribute to solutions for these types of problems in many areas of biomedical informatics.

## ACKNOWLEDGEMENTS

We thank S. Settnek and M. Connelly for their contributions to caBIO; J.-J. Maurer, L. Chatterjee, R. Chilukuri and P. Aggarwal for their work on the caDSR; M. Haber, L. Wright, J. Oberthaler and F. Rosenberg for contributions to EVS; D. Zimmerman for technical documentation; J. Silva, D. Warzel, B. Meadows and J. Abrams for their fundamental role in launching the CDE project. This work was supported by the National Cancer Institute, National Institutes of Health, U.S. Department of Health and Human Services.

## REFERENCES

- Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T. et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.
- Bechhofer,S. and Goble,C. (2001) Thesaurus construction through knowledge representation. *Data & Knowledge Eng.*, **37**, 25–45.
- Beck,K. (1999) *Extreme Programming Explained: Embrace Change*, 1st edn. Addison-Wesley, Boston, MA.
- Buetow,K.H., Klausner,R.D., Fine,H., Kaplan,R., Singer,D.S. and Strausberg,R.L. (2002) Cancer molecular analysis project: weaving a rich cancer research tapestry. *Cancer Cell*, **1**, 315–318.
- Dolin,R.H., Alschuler,L., Beebe,C., Biron,P.V., Boyer,S.L., Essin,D., Kimber,E., Lincoln,T. and Mattison,J.E. (2001) The HL7 clinical document architecture. *J. Am. Med. Inform. Assoc.*, **8**, 552–569.
- Dowell,R.D., Jokerst,R.M., Day,A., Eddy,S.R. and Stein,L. (2001) The distributed annotation system. *BMC Bioinformatics*, **2**, 7.
- Forrey,A.W., McDonald,C.J., DeMoor,G., Huff,S.M., Leavelle,D., Leland,D., Fiers,T., Charles,L., Griffin,B., Stalling,F. et al. (1996) Logical observation identifier names and codes (LOINC) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. *Clin. Chem.*, **42**, 81–90.
- Kent,W.J., Sugnet,C.W., Furey,T.S., Roskin,K.M., Pringle,T.H., Zahler,A.M. and Haussler,D. (2002) The human genome browser at UCSC. *Genome Res.*, **12**, 996–1006.
- Kruchten,P. (2000) *The Rational Unified Process: An Introduction*, 2nd edn. Addison-Wesley, Boston, MA.
- Lindberg,D.A., Humphreys,B.L. and McCray,A.T. (1993) The unified medical language system. *Methods Inf. Med.*, **32**, 281–291.
- McCray,A.T. and Nelson,S.J. (1995) The representation of meaning in the UMLS. *Methods Inf. Med.*, **34**, 193–201.
- Riggins,G.J. and Strausberg,R.L. (2001) Genome and genetic resources from the cancer genome anatomy project. *Hum. Mol. Genet.*, **10**, 663–667.
- Schaefer,C., Grouse,L., Buetow,K. and Strausberg,R.L. (2001) A new cancer genome anatomy project web resource for the community. *Cancer J.*, **7**, 52–60.
- Silva,J.S., Ball,M.J. and Douglas,J.V. (2001) The Cancer Informatics Infrastructure (CII): an architecture for translating clinical research into patient care. *Medinformatics*, **10**, 114–117.
- Stein,L. (2002) Creating a bioinformatics nation. *Nature*, **417**, 119–120.
- Strausberg,R.L., Greenhut,S.F., Grouse,L.H., Schaefer,C.F. and Buetow,K.H. (2001) *In silico* analysis of cancer through the Cancer Genome Anatomy Project. *Trends Cell Biol.*, **11**, S66–S71.
- Wang,L., Riethoven,J.J. and Robinson,A. (2002) XEMBL: distributing EMBL data in XML format. *Bioinformatics*, **18**, 1147–1148.