BIOINFORMATICS

Vol. 16 no. 5 2000
Pages 465–477

BIND—a data specification for storing and describing biomolecular interactions, molecular complexes and pathways

Gary D. Bader[1] and Christopher W. V. Hogue[2,*]

[1]Department of Biochemistry, University of Toronto/Samuel Lunenfeld Research Institute, Toronto, M5G 1X5, Canada and [2]Samuel Lunenfeld Research Institute, Toronto, M5G 1X5, Canada

Received on November 1, 1999; revised and accepted on March 1, 2000

Abstract

Motivation: *Proteomics is gearing up towards high-throughput methods for identifying and characterizing all of the proteins, protein domains and protein interactions in a cell and will eventually create more recorded biological information than the Human Genome Project. Each protein expressed in a cell can interact with various other proteins and molecules in the course of its function. A standard data specification is required that can describe and store this information in all its detail and allow efficient cross-platform transfer of data. A complete specification must be the basis for any database or tool for managing and analysing this information.*

Results: *We have defined a complete data specification in ASN.1 that can describe information about biomolecular interactions, complexes and pathways. Our group is using this data specification in our database, the Biomolecular Interaction Network Database (BIND). An interaction record is based on the interaction between two objects. An object can be a protein, DNA, RNA, ligand, molecular complex or an interaction. Interaction description encompasses cellular location, experimental conditions used to observe the interaction, conserved sequence, molecular location, chemical action, kinetics, thermodynamics, and chemical state. Molecular complexes are defined as collections of more than two interactions that form a complex, with extra descriptive information such as complex topology. Pathways are defined as collections of more than two interactions that form a pathway, with additional descriptive information such as cell cycle stage. A request for proposal of a human readable flat-file format that mirrors the BIND data specification is also tendered for interested parties.*

Availability: *The ASN.1 data specification for biomolecular interaction, molecular complex and pathway data is available at ftp://bioinfo.mshri.on.ca/pub/BIND/Spec/bind.asn. An interactive browser for this document is available through our homepage at http://bioinfo.mshri.on.ca/BIND/asn-browser/.*

Contact: *hogue@mshri.on.ca*

Introduction

Technological advances and mounting interest have pushed proteomics into the scientific spotlight. This growing field encompasses the study of proteins, both in structure and in function, contained in a proteome—the protein equivalent of a genome. Because of increased interest and technique automation (Mendelsohn and Brent, 1999), the rate of proteomic data production is growing in a similar fashion as that of genomics a decade ago. For example, mass spectrometers, gene chips, and two-hybrid systems have made cellular signaling pathway mapping faster and easier and consequently these are becoming large producers of data. Protein–protein interaction and more general biomolecule–biomolecule (protein–DNA, protein–RNA, protein–small molecule, etc.) interaction information is being generated and recorded in the literature. Lessons from the genomic era have taught us that large amounts of related data recorded in scientific journals soon becomes unmanageable. A well designed common data specification based on a model of the biological information is therefore required to describe and store biomolecular interaction data.

Any well designed data specification for the storage and management of biomolecular interaction and biochemical pathway data should possess certain properties.

1. It should be able to describe all of the details of the biological data, from simple binary interactions to large-scale molecular complexes and networks of pathways and interactions. It must be possible to store protein, DNA, RNA, and other molecules in full atomic detail, since character based sequence abstractions of biomolecules often miss important

*To whom correspondence should be addressed.

chemical features, such as methylation on DNA. This allows as much data as possible to be stored for scientific use in electronic form rather than in print.

2. It should be easily computable. A computer should be able to easily read, write and traverse the specification. This facilitates maintenance of a database of such information, creation of advanced queries and querying tools and development of computer programs that use the information for data visualization, data mining and visual data entry.
3. It should be platform and database independent. Tools written for one platform should be able to read data created on another platform directly. Any database management system should be able to handle the data structure without modification as well.
4. It should be succinct and easy for humans to understand. Field to data correspondence should be very clear and a human readable format of the specification should be available.

This paper describes a data specification for biomolecular interaction, molecular complex, and molecular pathway data that holds the above mentioned properties. It has been designed for a database called the Biomolecular Interaction Network Database (BIND) and has been written in a data specification language called Abstract Syntax Notation.1 (ASN.1) (http://www.oss.com/asn1/index.html). The US National Center for Biotechnology Information (NCBI) uses ASN.1 to describe and store all of its biological and publication data and all of GenBank, MMDB and PubMed (Ostell and Kans, 1998). BIND inherits the NCBI data model, which provides a solid foundation for the BIND data specification through the use of mature NCBI data types that describe sequence, three-dimensional structure and publication reference information.

Although the specification is written in ASN.1, it is important to realize that it is not restricted to this syntax. The data structures can be readily translated to other common data specification languages such as CORBA IDL (Object Management Group , 1996) or XML (http://www.w3.org/XML) if the need arises. Aside from ASN.1, no other biological data specification is sufficiently rich in mature data types to use as a foundation for BIND without first building and testing those base data types.

With the BIND data specification, we have tried to answer the question 'Can complex cellular pathway information be efficiently represented in a computer?' BIND defines three main data types: interactions, molecular complexes, and pathways. Each of these objects is composed of various component and descriptor objects that are either defined in the specification proper or inherited from the NCBI ASN.1 data specifications. For example, an interaction record contains, among other data objects, two BIND-objects. A BIND-object describes a molecule of any type and is itself defined using simpler sub-objects. Normally, a BIND-object describing a biopolymer sequence will store a simple link to a sequence database, such as GenBank (Benson *et al.*, 1999). If, however, the sequence is not present in a public database, it can be fully represented using an embedded NCBI-Bioseq object. The NCBI-Bioseq object is how NCBI stores all of the sequences in GenBank and is a mature data structure. BIND also inherits the NCBI taxonomy model (also used and supported by EMBL, DDBJ and Swiss-Prot) and data, via an inherited NCBI-BioSource, and is designed so that interactions can be both inter- and intra-organismal. Sequence, structure, publication, taxonomy and small molecule databases have provided a strong foundation for BIND.

The need for the BIND specification

It is important to design well thought out methods for the electronic management of complex biological data, such as molecular interactions, now, before the information becomes overbearing for any one expert. This scenario has already occurred with current resources containing biomolecular sequence information such as GenBank or SWISS-PROT (Bairoch and Apweiler, 1999). It is becoming apparent that the complexity of genomics may be overshadowed by the complexity of molecular and, in particular, protein interactions in the cell. Of the 60 000–100 000 anticipated human genes, only a small fraction encode classical 'enzymes', perhaps only a few thousand. It is probable that most of the proteins encoded in the human genome are large, multi-domain molecules that participate in molecular interactions with other proteins, DNA, carbohydrates and small molecules. Thus, it is not unreasonable to say that there are more protein–protein interactions than sequences (Marcotte *et al.*, 1999).

Other interaction databases have been developed such as DIP (Marcotte *et al.*, 1999), BRITE (http://www.genome.ad.jp/brite/), CSNDB (Igarashi and Kaminuma, 1997) and Interact (Eilbeck *et al.*, 1999). Of these efforts, none are general for all biological molecular interactions and all lack a data specification that can handle the complexity and scale of the anticipated data. Even the GenBank/EMBL (Stoesser *et al.*, 1999) DDBJ (Sugawara *et al.*, 1999) feature table (DDBJ/EMBL/GenBank, 1997) contains space for recording interactions. Certain keys such as `"misc_binding"` allow a sequence submitter to enter and maintain interaction information within sequence records. Other standard feature table keys to indicate binding events are the `"protein_bind"` key used to annotate non-covalent protein binding sites on nucleic acid sequences, and the `"RBS"` key used to

indicate a ribosome binding site. Each of these feature table entries has only one single mandatory qualifier, `/bound_moiety="text"`, that allow the user to describe in plain text the bound moiety. There are other optional qualifiers that include `"citation"`, `"db_xref"`, and a series of free text fields that can be used to enter completely unformatted text data.

One problem in using these feature keys within sequence records is that this part of the specification is not suited to generate machine-readable information necessary to allow computer programs or individuals to explore the vast information space of interactions. Larger problems with the feature table are that it is DNA centric and thus poor for protein annotations and it does not fully represent the richness of the NCBI ASN.1 specification. The feature tables are also underutilized by sequence depositors as demonstrated for *Drosophila melanogaster* (Mohr *et al.*, 1998). Features as described by GenBank/EMBL/DDBJ are not sufficient and not widely used, and we should not expect them to be used, to capture molecular interaction information.

The BIND data model

This section describes the three main types of data objects in the BIND specification—interaction, molecular complex and pathway—as well as useful database management and data exchange objects. Explanations of the various objects in the specification are given along with examples. The specification will be explained as if it were being used to describe a single record in a database. The specification is available via FTP from ftp://bioinfo.mshri.on.ca/pub/BIND/Spec/bind.asn.

It is suggested that the reader follow the specification along with this paper. The data model is shown in Figures 1, 2 and 3 using Unified Modeling Language (UML: see http://www.rational.com/uml). Wherever possible, this specification is meant to reference information from other databases rather than storing the information as a copy. This avoids unnecessary duplication of information among databases and helps maintain data integrity (if the information in a referenced record in one database is updated, the other databases that reference the record are all automatically updated). All fields are non-optional unless stated otherwise.

A BIND-object

A BIND-object represents any chemical object—atom, molecule or complex of molecules. A BIND-object contains the following items.

1. *A short-label field to contain a short name for a molecule*. For example, ATP, IP3, S4 and HSP70 are acceptable short labels for ligands and proteins, respectively. Having a non-optional short label ensures that at least some descriptive data is entered for a molecule. This information is also useful to construct top-level descriptions regarding a particular record. For example, a simple description of an interaction between two proteins can be constructed using the short labels of the two BIND-objects in an interaction record. A graphical view of an interaction would be labeled with the short-label field.

2. *A BIND-object-type-id object to contain the type of the molecule and a reference to another database containing a record for that molecule*. In this way, for instance, large DNA records are referenced rather than duplicated. A molecule type may be 'not-specified', 'protein', 'dna', 'rna', 'ligand', 'interaction' or 'molecular complex'. Molecules of unknown type may be stored by specifying the type of molecule as 'not-specified'. This type requires no further data input.

Protein, DNA and RNA all require a BIND-id object. This object can store accession numbers to any other database. It has special fields 'gi' or Geninfo and 'di' or domain identifier for the NCBI Entrez system (Schuler *et al.*, 1996) and a database of domains under development at the Samuel Lunenfeld Research Institute, respectively. Any other accession number or numbers/strings to reference records in other databases can be stored in a set of NCBI Seq-ids present in the data object. All fields in BIND-id are optional so molecules stored internally in a BIND record that are not present in other databases (and so do not have accession numbers) can be properly saved.

Molecules of type 'ligand' require a BIND-ligand-id object. This object can contain a reference to an internal small molecule database or any other small molecule database via a database name and an integer and/or character based accession number.

BIND-objects of type 'complex' require an integer accession number to a BIND molecular complex record.

3. *A BIND-object-origin data structure*. This structure contains a choice of origin between 'not-specified', 'org' or organismal, and 'chem' or chemical. BIND-objects of unknown origin would have origin type 'not-specified'. Chemical objects that are derived directly from organisms, such as DNA, would be specified to be origin type 'org' and are required to be associated with an NCBI BioSource object. A BioSource object can contain much descriptive data about an organism and the biological source of a compound. It also contains a reference to a taxonomy database. This information can be entered automatically if a GI is known for a biological sequence molecule, since a BioSource is part of the NCBI Bioseq object which stores biological sequences in Entrez. If a GI is not given, a BioSource can be created.

Molecules derived purely from chemical means are of origin type 'chem' and require a BIND-chemsource object. The BIND-chemsource object contains a set of

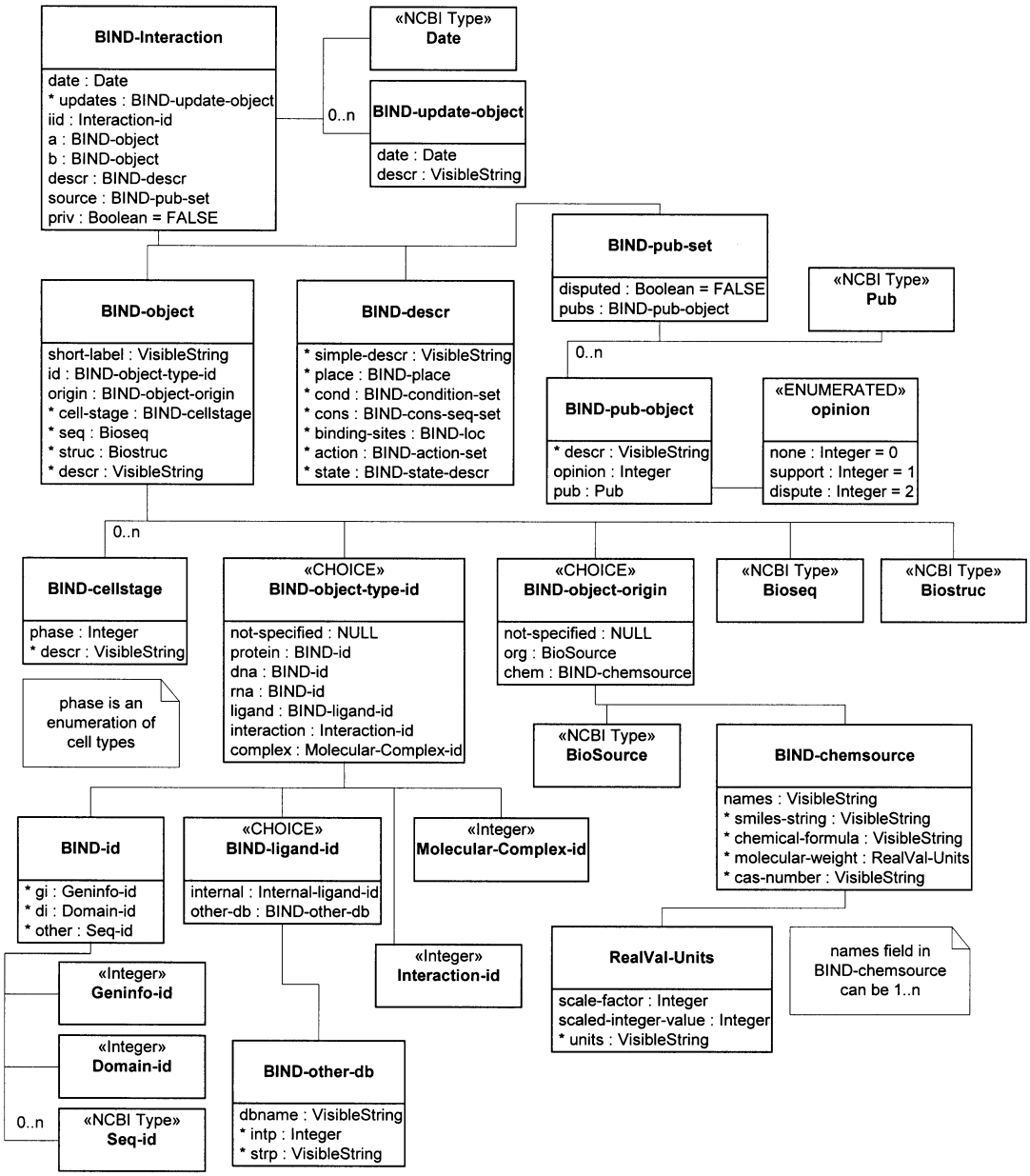

Fig. 1. Graphical representation of the BIND data model in UML. This figure expands upon sub-types of BIND-Interaction except for BIND-descr, which is shown in Figure 2. Data fields preceded by an asterisk are optional in the specification. Short ASN.1 'ENUMERATED' lists in are shown in full, while long lists are only described in the specification and referenced using a UML note. ASN.1 'CHOICE' elements are marked in the figure. Referenced NCBI data types are marked 'NCBI Type' and are not expanded. See the NCBI data model for further details on those types. Integer types are marked as such.

names for the chemical, usually a common name and any synonyms, a SMILES string (Weininger, 1988), the chemical formula, molecular weight (a RealVal-Units object), and a CAS registry number (http://www.cas.org). A SMILES string is a standard way of representing a molecule's structure using ASCII characters. Many chemistry computer applications are available to manipulate and use data of this type. Three-dimensional structure of a molecule can be predicted from a SMILES string to a high degree of accuracy using commercial chemistry applications such as Corina (Gasteiger, 1996) and others. A CAS number is a reference number to the information regarding a chemical compound in the Chemical Abstracts Service. This service contains data on 22 914 327

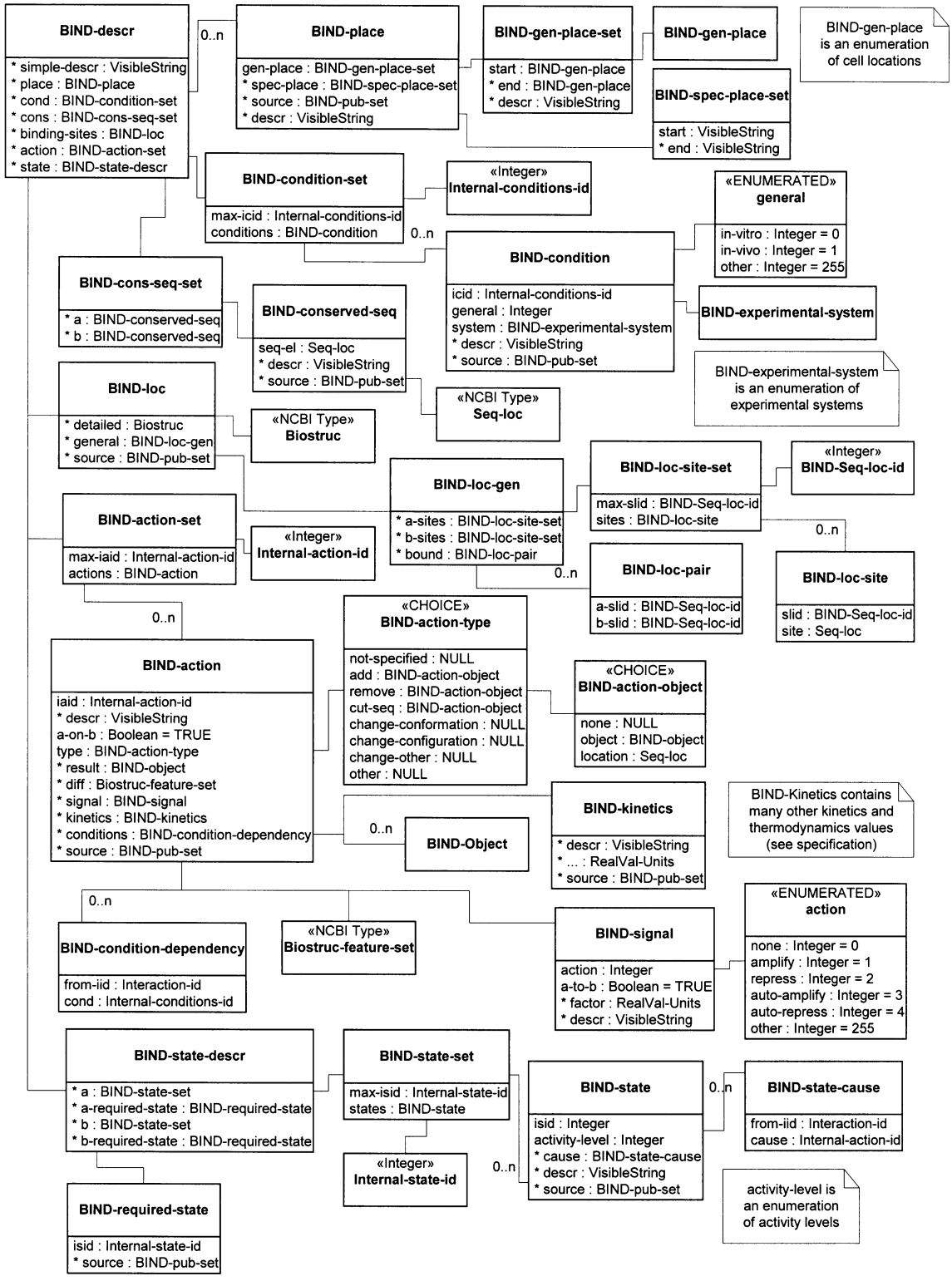

Fig. 2. Continued UML representation of the BIND data model, showing the BIND-descr type and its sub-types. See Figure 1 caption for notation explanation.

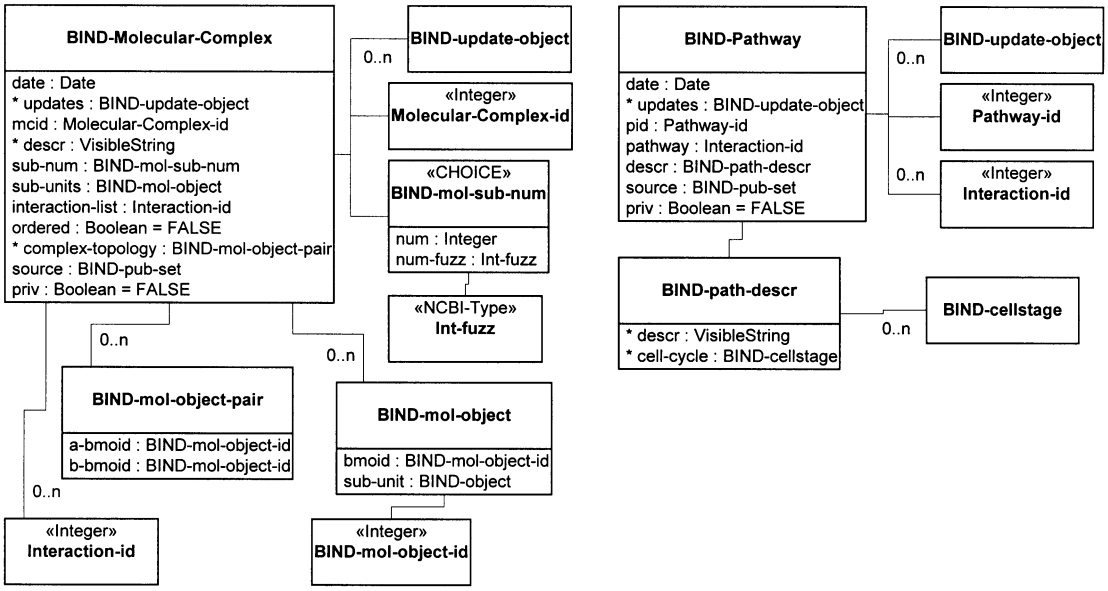

Fig. 3. Continued UML representation of the BIND data model, showing the BIND-Molecular-Complex and BIND-Pathway data types. See Figure 1 caption for notation explanation.

chemical compounds (as of February 24, 2000). Of all the fields in a BIND-chemsource object, only 'names' is non-optional. This means that for a BIND-object to be declared a ligand of chemical origin, one must only provide a pointer to a small molecule database and one name of the chemical.

4. *An optional BIND-cellstage list to contain a list of cell cycle stages in which this object is found, or expressed, in the given organism*. This information is only relevant for BIND-objects of organismal origin. A BIND-cellstage object is an enumeration of all of the basic cell stages in the cell cycle. It contains an optional text description field that can describe other cell stages that are not present in the enumeration.

5. *An optional NCBI Bioseq object to store a biological sequence if a record for the sequence is not present in any public database*. The Bioseq may also be used to store the experimental form, such as His tagged proteins or mutants, of the biological sequence if it is different from any public database record. This field is only relevant for biological sequences. Bioseqs can be prepared using Sequin (Kans and Ouellette, 1998) and can be exchanged with NCBI.

6. *An optional NCBI Biostruc object to store a three-dimensional atomic structure of any chemical object, from an atom to a complex of molecules, if the data is not present in any public database*. The Biostruc specification allows a chemical graph to be stored without coordinates. This is most useful for storing small molecule structures or post-translationally modified forms of a biomolecule. Thus, chemical entities within a BIND object can be described in precise detail.

The presence of these powerful and mature data structures in this part of the specification signifies that BIND is not completely reliant on other databases. Most of the information present in any public sequence or three-dimensional molecular structure database can be stored using the BIND specification if necessary.

7. *An optional free flow text description of the BIND-object*. This field could contain, for example, a full name for a molecule such as Adenosine Triphosphate (ATP).

BIND-Interaction

The BIND-Interaction object is the fundamental component for storing data in this specification. It defines and describes the interaction between any two molecules, or even atoms. The majority of the information that can be stored is, however, used to describe interactions between proteins, DNA and RNA. We will only refer to interactions between molecules rather than between molecules and atoms from this point on.

An interaction contains a NCBI Date object, a sequence of updates for an audit trail, an Interaction Identifier (IID) accession number, two interacting molecules (BIND-object), a description of the interaction, a series of publications and a private flag. We plan to control the BIND IID number space using a unique key server. Molecule A binds to molecule B and both are stored using BIND-objects (described above).

The BIND-descr object stores most of the information in an interaction object. It contains text description of the interaction, information on the cellular place of interaction, experimental conditions used to observe the interaction, conserved sequence comment of molecules A and/or B if they are biological sequences, location of binding sites on molecules A and B, chemical actions mediated by the interaction and chemical states of the molecules A and B.

A BIND-pub-set is included to store empirical evidence references, usually publications, that 'support', 'dispute' or have 'no opinion' regarding the actual interaction. The dispute flag allows the database to track experimental trends and offer a machine-readable way to find discrepancies or differences of opinion.

Finally, a private flag which defaults to FALSE is included in an interaction record. The flag indicates whether or not to export this record during a data exchange procedure. In a public database, a private record is not available to the public. This may be because a record has not been completed or information in the record has not been verified. In a private database, the private flag means that the record could be viewed internally, but it would never be exported. In this situation, a private record might contain proprietary information and the database may contain a mix of these and public records imported from a public database.

Interaction description—BIND-descr

All of the objects directly linked in this structure are optional to allow any level of richness of data to be stored. BIND-descr contains the following items.

1. *A simple text description of the interaction*. This free flow text is meant to be a short description of the interaction such as, 'transcription factor *X* binds to a region of human DNA in section *x* of chromosome 11'.

2. *A sequence of BIND-place objects*. A BIND-place object stores information about the location of the interaction with respect to the cell. The place of an interaction is meant to be the location where molecules A and B come together in a biologically meaningful way. This object contains a BIND-gen-place-set object for storing general place data, an optional BIND-spec-place-set object for storing specific place data, an optional BIND-pub-set for storing publications referring to the localization of an interaction, and an optional text description field. A BIND-gen-place-set contains a start and an optional end place for the interactions, specified by an enumerated list of general places in the cell. Storing a start and an end place for an interaction takes into account the possibility of an interaction translocating across membranes and ending up in different sub-cellular compartments. The general enumeration of cell places allows a computer to understand the location of the interaction. Only basic cell places are present in the list. This is important for data visualization programs that need to be able to draw molecules in the correct places on a diagram of a cell. A human readable description of cellular place can be stored in the BIND-spec-place-set. This object contains a text description of a start and an optional end place for an interaction. More specific data regarding the location of interaction, such as in what part of a membrane, apical or basal, an interaction occurs can be stored in the BIND-spec-place-set object.

Multiple BIND-place objects are present to allow storage of an interaction that may be present only at certain separate places within and around the cell. More than one place object can also be used to describe an interaction occurring between two molecules over multiple sub-cellular compartments, as might be the case for transmembrane receptor proteins with large extra- and intra-cellular domains.

3. *A BIND-condition-set to store a list of experimental conditions used to observe the interaction*. Experimental conditions information stored should be sufficient to allow recreation of the original experiment. An experimental condition is described using a BIND-condition object. This object contains an Internal-conditions-id (ICID) number which can be used to reference a particular experimental condition in the BIND-condition-set. A general experimental condition is an enumeration of three general conditions, *in-vitro*, *in-vivo* and other. A BIND-experimental-system object is present and is an enumeration of most popular experimental techniques, with 34 techniques listed in the specification. This field has been simply declared as an INTEGER enumeration type so that it can be easily extended with new experimental systems as they become available. Declaring a type as INTEGER in ASN.1 instead of enumeration prevents generated code from checking the name of the enumerated value against the specification. This means that items may be added to the list at a later date without disrupting tools that are based on previous specifications. A BIND-condition object also contains a free human readable text description. This field could be used to describe a system further or could be used to name a system if 'other' has been specified as the BIND-experimental-system object. A BIND-pub-set is also provided in order to store publications related to the experimental systems described in the BIND-condition object.

4. *A BIND-cons-seq-set to store information about an evolutionarily conserved sequence if either molecule A or molecule B is a biological sequence*. This information is simply meant to be a comment on the possible importance of certain sequence elements that have been noticed to be conserved via phylogenetic or other evolutionary analysis. It is possible that information about the conserved sequence is known for molecules in an interaction that is not very well characterized. This data might be useful to in-

vestigators interested in further studying the interaction. A BIND-cons-seq-set contains conserved sequence information about molecules A and B in a BIND-conserved-seq object. Semantically, a BIND-conserved-seq object may only be instantiated with data if the molecule that it refers to is a biological sequence. A BIND-conserved-seq object contains an NCBI Seq-loc object. A Seq-loc can contain a location or a set of locations for any linearly numbered biological sequence. A free text description is also included in a BIND-conserved-seq. It is suggested that the method of determining the conserved sequence, for example a phylogenetic tree program such as PHYLIP (http://evolution.genetics.washington.edu/phylip.html) or an alignment program such as PSI-BLAST (Altschul *et al.*, 1997) or CLUSTAL (Higgins *et al.*, 1996), be stored in the 'descr' field. A BIND-pub-set object is provided to store publications pertaining to a conserved sequence comment.

5. *A BIND-loc to store binding site information*. A BIND-loc can store three-dimensional atomic level detail of an interaction site using an NCBI Biostruc. A BIND-loc-gen object is present to store binding sites in an interaction at the sequence element level of detail. Therefore, only interactions involving biological sequences can hold general binding site information. The BIND-loc object also includes a BIND-pub-set for storing publications related to the binding site. All top level fields are optional allowing detailed, general and/or source information to be represented. Expanding further, the BIND-loc-gen object contains a list of binding sites on molecule A and a list of binding sites on molecule B. This information is contained in a BIND-loc-site-set object which contains a sequence of binding sites defined in BIND-loc-site objects. Each BIND-loc-site element contains an NCBI Seq-loc element and an internal reference integer ID called a BIND-Seq-loc-id. Since each binding site is numbered in a BIND-loc-site-set, it can be referenced by other objects.

A BIND-loc-gen object also contains an optional BIND-loc-pair object which specifies which binding sites on A bind to which binding sites on B. The binding sites are referenced from the BIND-loc-site-set objects, so in order to use a BIND-loc-pair object binding sites on molecules A and B must already be defined. This simple binary mapping allows most experimental binding information, such as that generated from footprinting analysis, to be stored.

6. *A BIND-action-set to describe the chemical action(s) mediated by this interaction*. A set of actions is required because there are many examples of interactions having multiple chemical actions. For instance, a kinase may phosphorylate a protein more than once in separate chemical actions or a restriction enzyme may cleave a molecule of DNA in more than one place. A BIND-action-set contains a set of elaborate BIND-action objects. Each BIND-action object in a set is numbered with an Internal-action-id (IAID) integer so that it can be referenced by other data types.

A BIND-action object contains an IAID number, an optional text description field for free flow text description of the chemical action and an optional BIND-pub-set for storing publications pertaining to this chemical action. A Boolean flag is included to specify the direction of the chemical action. If a-on-b is set to true, then molecule A acts on molecule B, and vice versa. This value defaults to true. The type of action is defined in the BIND-action-type object. The BIND-action-type object is a choice element that stores the type of chemical action and an associated data object. The possible choices of actions are 'not-specified' for an unknown chemical action type, 'add' for adding a chemical object, 'remove' for removing a chemical object, 'cut-seq' for a cut in a biological sequence, 'change-conformation' for a change in conformation, 'change-configuration' for a change in configuration, e.g. by an epimerase or isomerase, 'change-other' for another type of change, such as a metal ion exchange, and 'other' for any other chemical action. Types 'add', 'remove' and 'cut-seq' are associated with a BIND-action-object to store related data. A BIND-action-object is a choice element that can store nothing, with a choice of NULL, a BIND-object, or a site on a sequence using a Seq-loc. The 'object' choice of the BIND-action-object is only relevant for the 'add' and 'remove' choices of the BIND-action-type. The BIND-object is meant to store a description of the chemical compound that is added or removed. An example would be a phosphate group that could be added by a kinase enzyme or removed by a phosphorylase enzyme. The 'location' choice of the BIND-action-object is only relevant for the 'cut-seq' choice of the BIND-action-type. The Seq-loc is meant to store the position(s) where a biological sequence is cut. An example would be the locations after which a restriction enzyme cuts DNA or the sites after which a protease cleaves in a protein. The choice of 'none' can be used for either 'add', 'remove' or 'cut-seq' if information that would otherwise be stored is not known.

Continuing with the description of the BIND-action object, an optional result field is present as a sequence of BIND-objects to store the resulting molecule(s) from a chemical action. For instance, if a molecule of DNA was methylated, the description of the methylated DNA could be stored in a BIND-object. If a protein molecule was cut at various locations, all resulting protein molecule fragments could be described with the BIND-object sequence. With a sequence of interacting proteins where A binds to B, B binds to C, etc., the result field storing the full chemical form of B in the A–B interaction, for example, could be used directly in the B–C interaction record. This allows the exact description of sequential

chemical modifications on a biological sequence that would otherwise not be possible given the standard sequence representation alone.

A Biostruc-feature-set that can contain residue or atomic level of detail differences in a molecule created by this chemical action is also present. The molecule that is different in this case is based on the direction of the chemical action. If the direction is molecule A to molecule B, any information stored in the diff field would pertain to molecule B, not molecule A. This field allows even small changes to molecules to be represented, as in the example of a chemical action reducing a double bond by adding two hydrogen atoms across it. The addition of the two hydrogen atoms could be recorded as differences on an atomic structure. This information requires the presence of atomic level detail data for the molecule being changed. The diff field can also represent changes made to the substrate of the chemical action. In an example of a phosphate added to a protein on a specific tyrosine residue by a phosphokinase enzyme, the diff field would simply be the position in the protein sequence of the tyrosine that was being changed.

An optional BIND-signal object is included in the BIND-action object to store directional information related to chemical signal as it is found in cell signaling pathways. This data is really a more general notion of kinetics describing the signal transduction. The signal could, for example, be the activation of proteins in a signaling cascade via phosphorylation such as in a MAP kinase pathway. BIND-signal contains an enumerated type describing the signal modification from a top-level viewpoint. Possible values are 'none', 'amplify', 'repress', 'auto-amplify', 'auto-repress', and 'other'. The direction of the signal is stored in the a-to-b Boolean flag, which defaults to true. If a-to-b is true, the direction of the signal is from molecule A to molecule B and vice versa. An optional RealVal-Units object of the 'factor' field can store the factor of signal amplification or repression if they occur. Signal amplification in the cell is really just the recruitment of molecules one step further down in the pathway by the molecule at the current step. So, if molecule A activates molecule B by removing a phosphate in a signaling pathway and there is amplification at this step, in the cell, molecule A activates many molecules of B causing a strengthening of the chemical signal by a measurable factor that may be stored. An optional free text description is available in the BIND-signal object as well. This field should contain some description of the signal action if 'other' is specified in the 'action' field.

Kinetic and thermodynamic data may also be optionally stored in the BIND-action object using the BIND-kinetics object. The BIND-kinetics object offers specified real value and text description fields for common kinetics (e.g. Michaelis–Menten) and thermodynamic values as well as providing a sequence of BIND-kinetics-other objects to store any other text or real number values that may be pertinent. A BIND-pub-set object is also present to store publications that relate to any of the information stored. All objects in the BIND-kinetics object are optional to allow any combination of values to be stored.

Also in the BIND-action object, a link to a sequence of experimental conditions used to observe this chemical action is optionally provided using a sequence of BIND-condition-dependency objects. The BIND-condition-dependency object references a previously defined experimental condition by Interaction-id and Internal-conditions-id number. In this way, any experimental condition in a database using this specification may be uniquely referenced.

7. *A BIND-state-descr object for storing information on the chemical state of molecule A or molecule B.* The BIND-state-descr object stores a list of possible chemical states for molecules A and B in BIND-state-set objects as well as references to defined chemical states of A and B that are required for the interaction to take place, in BIND-required-state objects. More than one possible state can be saved because certain molecules can assume multiple states. One example is a protein enzyme which may be multiply phosphorylated to bring about different enzymatic activity levels, depending on the phosphorylation level. All fields in the BIND-state-descr object are optional allowing any combination of data objects to be stored. A BIND-state-set contains a sequence of BIND-state objects each numbered by an Internal-state-id (ISID) integer. A BIND-state object contains an optional enumerated list describing the general activity of the molecule, an optional sequence of BIND-state-cause objects, optional free text description and an optional BIND-pub-set for storing publications related to this chemical state description. The 'activity-level' list is a simple description and is purely subjective, but is still useful for discriminating various states of different activity, especially by a data visualization program which could colour molecules based on this information. The BIND-state-cause object can be used to uniquely reference previously defined chemical actions from this or other interactions that bring about this state. It contains an IID and an IAID. This functionality is very important in the specification because it allows full chemistry to be described when chemical actions and chemical states are taken together. Full chemistry means that all substrates, enzymes, products, bio-processed compounds etc. may be represented in full atomic level detail for all steps in a pathway. A certain chemical action can have a result (in the 'result' field of a BIND-action object) and a certain chemical state can reference the action that occurred to create it. In this way bi-directional linked lists can form networks that represent true chemical networks in a cell.

A molecular complex—BIND-molecular-complex

The BIND-Molecular-Complex object is the second of three top-level biological objects in the BIND specification. It is meant to store a collection of more than two interactions that form a complex, i.e. three or more BIND-objects that can operate as a unit. In this way, it is useful to store knowledge of molecular complexes and for use as a shorthand when defining interactions and pathways (see BIND-pathway).

A BIND-Molecular-Complex object contains similar administrative information fields as a BIND-Interaction. A Molecular-Complex-id (MCID) integer accession number is stored to uniquely identify molecular complexes. A BIND-pub-set is present to store publications that concern this molecular complex and a private flag is provided to mark this record as private using the same rules as the private flag of the interaction record.

Six other fields in the molecular complex store data directly relating to the complex. The 'descr' field optionally provides space for a human readable free text description of the molecular complex. The 'sub-num' field contains a BIND-mol-sub-num object that stores the number of sub-units (BIND-objects) in the molecular complex. The sub-unit number is a choice of an exact integer using the 'num' field or a fuzzy integer in the 'num-fuzz' field. The fuzzy number is stored using an NCBI Int-fuzz object which can store a number in a range, plus or minus a fixed or percentage amount, or store a set of alternatives for the number. Using a fuzzy number, complexes can be stored even when the exact number of sub-units is not known. Examples of such complexes are actin filaments or other parts of the cytoskeleton and virus coat proteins, both of which typically form using repeated units of a certain protein. Continuing with the BIND-Molecular-Complex, the 'sub-units' field can store the actual sub-units of the complex as a sequence of BIND-mol-object data types. The BIND-mol-object is simply a wrapper for a BIND-object that allows the BIND-object to be numbered using a BIND-mol-object-id integer (BMOID). Numbering the sub-unit BIND-objects allows the BIND-mol-object-pair to reference them for topology, as discussed below. The core component of the BIND-Molecular-Complex is the list of Interaction-ids which references previously defined interactions in a database. This means that most of the data for function, state, location, etc. for a molecular complex is actually stored in BIND-Interaction objects. This avoids some duplication of information. A Boolean flag marks the interaction list as being ordered or not. This should be true if the temporal order of interactions that form the complex is known and the IID list is ordered in that way. Ordering of sub-unit binding for some well studied biological complexes, such as the ribosome, is known.

An optional sequence of BIND-mol-object-pair objects is present in the BIND-Molecular-Complex and is meant to store a two-dimensional topology of the molecular complex. A BIND-mol-object-pair simply records a connected pair of BIND-mol-objects in the molecular complex by making a reference to two BMOID numbers of the sub-units that are connected. Together the BIND-mol-objects, as nodes, and the BIND-mol-object-pairs, as edges can describe the computer science concept of a graph. The topology information can allow a data visualization program to draw a representation of the actual shape of the complex.

Because most of the data for complexes is referenced from interaction records, a certain amount of automatic data entry can be used. A list of sub-units and the number of sub-units can be automatically entered by fetching the data from the given list of interaction records.

It can also be noted that a molecular complex can be defined if the pairwise interactions of which it is composed are not completely known. This can be done by creating a set of interaction objects with molecule A as a sub-unit of the complex and molecule B as 'not-specified'. This is useful since many preliminary studies of a molecular complex observe only that certain molecules interact, e.g. from gel data, but not how they interact.

A pathway—BIND-pathway

The final top-level biological object in the BIND specification is the BIND-pathway data type. It describes a collection of more than two interactions that form a pathway, i.e. three or more BIND-objects that are generally free from each other, but can form a network of interactions. Common examples include metabolic pathways and cell signaling pathways.

A BIND-Pathway object contains similar administrative information fields to a BIND-Interaction and a BIND-Molecular-Complex. Two other fields in the BIND-pathway object store information describing the pathway. A sequence of Interaction-ids that reference previously defined interactions that make up this pathway is stored. Extra descriptive information regarding the pathway is stored using a BIND-path-descr object. This object can optionally store free text describing the pathway and an optional sequence of BIND-cellstage objects that represent the phases of the cell cycle in which this pathway is in effect. Parts of the pathway may be constitutively present in the cell, while other parts that complete the pathway and allow activation may only be expressed at certain times during the cell cycle.

Other BIND ASN.1 objects

Publication set. A BIND-pub-set is used to hold all publications in BIND. It contains a list of BIND-pub-objects and a dispute flag. A BIND-pub-object contains an optional free text description of the publication, an enumerated opinion of the publication field and a NCBI

Pub object. The description field may hold any text data pertaining to the publication referenced by this object. The opinion field may hold the values: 'none', 'support' and 'dispute'. It is meant to convey the general opinion of the referenced publication in regard to the information in the ASN.1 object that contains the BIND-pub-set. The NCBI Pub object is used to store most of the data in PubMed and can represent almost any publication. It should be used to store a reference to PubMed whenever possible using either a Medline Unique Identifier (MUID) or a PubMed unique identifier (PMID).

Record update. If a record is updated in BIND, a description of the update should be added to a BIND-update-object. This object contains a NCBI Date object and a text description field. The description field may contain any information that a database implementation decides to store, but it should be complete and stored in a standard and automatic way within each implementation so that it can be easily parsed. Any information may be stored up to and including the entire previous record in ASN.1 value notation. This data is not meant to be human entered but rather maintained as a machine generated audit trail of any changes made.

Data exchange and data cross-referencing. Data exchange systems and database management data structures have been included in the specification as powerful tools to make implementations more robust.

BIND-Submit is the top-level object for data exchange while the cross referencing system involves many separate top-level data objects.

Data exchange—BIND-submit. The BIND-Submit object can be used to exchange any number of the top-level data types in the BIND specification, BIND-Interaction, BIND-Molecular-Complex, and/or BIND-Pathway objects. BIND-Submit stores an NCBI Date object, an optional BIND-Database-Site, a BIND-Submitter object, an optional BIND-Submit-id integer for identifying this submission, and fields for optionally storing BIND-Interaction-set, BIND-Complex-set, and BIND-Pathway-set objects.

A BIND-Database-site is a description of a database site. This object could be used if data was being submitted to BIND from any other database. It contains a free text description of the database site, usually the database name. Also present is a text field for database country of origin and an optional field used to store the World Wide Web Universal Resource Locator (WWW URL) of the homepage of the database on the Internet. An optional NCBI Pub object can store a Medline reference for this database.

A BIND-Submitter object contains information about a submitter to a BIND database. BIND-Submitter stores a BIND-Contact-info object which contains information about a person. A 'hold until published' Boolean flag is present which defaults to false to allow data submission prior to publication. Also present is an optional enumeration of possible submission types, either 'new', 'update', 'revision', or 'other'. An update is a change by an author while a revision is a non-author update. A free text field, 'tool', stores the name and version of the tool used to submit the record.

Personal contact information should be kept separate from BIND records to keep the submitter and ownership information anonymous and protected from improper use.

Actual records are stored in the BIND-Submit object in data set data types. The BIND-Interaction-set, BIND-Complex-set and BIND-Pathway-set are all present in the BIND-Submit object and are analogous in that they optionally store the date on which the set was collected, optionally the database from which the record set originates using a BIND-Database-site, and the respective sequence of records.

Cross-referencing the data

Since the BIND specification describes biological data from interactions to pathways and networks of pathways, the information space represented resembles a largely undirected graph with molecules as vertices and their interactions as edges. Cross-referencing information allows the graph to be easily traversed using simple indexed lookup techniques. If cross-referencing were not used in a system such as this, all records would have to be examined at each traversal of the data space. Instead of creating traditional large, unwieldy indexes and tables to speed the traversal process, ASN.1 objects are directly specified to store cross-reference information. This represents an object oriented database index system. Each BIND database accession number as well as NCBI GI, MUID and PMID and SLRI DI accession numbers has its own associated cross-reference object. This information may be easily exported and used by other databases to link their sequence or structure data back to BIND.

When updating cross-reference information, only one level of the graph is traversed, so as not to make the index overly complicated. Any time one of the three top level objects is created that contains a cross-referenced accession number, the BIND-Cross-Ref object lists are updated. In this way, any search using a cross-referenced accession number instantly retrieves all of the interaction, complex and pathway records that contain it.

The interaction cross-reference data is stored in a BIND-Iid-Cross-Ref object. This data type contains the IID of the interaction being cross-referenced in this object. The 'iids', 'pids' and 'mcids' fields contain a list of IIDs, PIDs and MCIDs, respectively of interactions, pathways and complexes that contain this interaction. A BIND-

Submitter object is included to privately store submitter information for every interaction.

Molecular complex cross-reference information is stored in a BIND-Mcid-Cross-Ref object which is completely analogous to the BIND-Iid-Cross-Ref object.

Pathway cross-reference data is contained in a BIND-Pid-Cross-Ref object. This object only keeps a list of submitters for each pathway record. Since no other objects can reference a pathway record, the BIND-Pid-Cross-Ref object does not contain references to other records.

The GI/DI cross-reference information is stored in a BIND-Cross-Ref object. This object links a biological sequence to a list of interactions, molecular complexes and pathways that contain it.

PMID/MUID cross-reference data is maintained in a BIND-Pub-Cross-Ref object. This cross-reference scheme is analogous to that of GI/DI accession numbers.

The full cross-reference system allows quick and easy searching of the database by any of the five indexed accession numbers.

Exported data types

Typical ASN.1 data specifications make certain data types available for use by other ASN.1 specifications by exporting them. BIND currently exports the top-level data types BIND-Submit, BIND-Interaction, BIND-Interaction-set, BIND-Pathway, BIND-Pathway-set, BIND-Molecular-Complex and BIND-Complex-set.

Flat-file record format

Many current biological data specifications are available in a flat-file format for use with simple flat-file databases. Examples include the GenBank flat-file format and the FASTA format for biological sequence representation. The BIND specification is not currently available in a flat-file format, in part because flat-file formats do not exist for all the embedded data types (e.g. Biostruc). A request for proposal (RFP) is hereby issued for a flat-file database record format that mirrors the BIND ASN.1 specification. Scientific and academic groups interested in the availability of such a format should consider constructing a proposal for submission to the BIND group. It is hoped that, given the demand for a flat-file representation, enough proposals will be collected from the community at large to allow the managed emergence of a common and compatible format. Once a flat-file representation has been settled upon, the BIND group will build the necessary ASN.1⇔flat-file conversion software tools to make it useful.

Implementation

This section gives an overview of the BIND database. The BIND database may be accessed from the web page http://bioinfo.mshri.on.ca. The implementation allows data entry and data retrieval supporting the full BIND 1.0 ASN.1 specification. Programmed fully using the C programming language for maximum speed and compatibility, the BIND application programming interface (API) has been written to allow applications to easily use data in the BIND database. The API makes use of two C libraries, the NCBI Toolkit (ftp://ncbi.nlm.nih.gov/toolbox) for ASN.1 handling and more and the CodeBase (http://www.sequiter.com/) database library for a database implementation. Using this API, web-based applications have been developed for data entry, retrieval and management. All data is entered and retrieved using web-based forms generated by CGI programs written in C. Interaction data is currently being entered using this web-based user interface and the system is constantly being updated with the help of user feedback.

The BIND API will eventually be made available to the public through our FTP site and will support remote data access via our own HTTP servers.

The BIND database uses the Seqhound database system as a resource (Michalickova and Hogue, unpublished data). Seqhound is our own in-house mirror of GenBank, the NCBI taxonomy database and the PDB (Bernstein *et al.*, 1978) data in NCBI MMDB form (Hogue *et al.*, 1996). Seqhound derived data allows BIND to quickly and easily use sequence, taxonomy and three-dimensional molecular structure information for validation and for information retrieval.

Future work

The data specification is under constant examination, since it is already being used in our own implementation. As time passes, the process of modifying the specification will yield mature and stable data types. We welcome feedback from anyone using the BIND database or specification. Data visualization and data mining systems have been designed but not written for the database implementation.

Conclusion

We have presented a data specification for a standard way of representing biomolecular interaction, molecular complex and pathway information using the internationally standard ASN.1 syntax. The need for such a representation is paramount at this time as the scientific community, and specifically the proteomics community, gears up for an explosion of interaction and pathway data.

We encourage the use of and comments on this data specification and the related software tools that we will provide and maintain. Data specifications require community input in order to mature and become useful.

Acknowledgements

We would like to thank our colleagues at the Samuel Lunenfeld Research Institute for their support in our work, especially Katerina Michalickova for developing the Seqhound database system. Moez Dharsee, Ian Donaldson and Francis Ouellette were helpful in looking over the manuscript. We are also grateful to the authors of the original NCBI ASN.1 data specifications, James Ostell, Steve Bryant, Hitomi Ohkawa and others, for making our task easier.

References

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Bairoch,A. and Apweiler,R. (1999) The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Res.*, **27**, 49–54.

Benson,D.A., Boguski,M.S., Lipman,D.J., Ostell,J., Ouellette,B.F., Rapp,B.A. and Wheeler,D.L. (1999) GenBank. *Nucleic Acids Res.*, **27**, 12–17.

Bernstein,F.C., Koetzle,T.F., Williams,G.J., Meyer,E.F.J., Brice,M.D., Rodgers,J.R., Kennard,O., Shimanouchi,T. and Tasumi,M. (1978) The protein data bank: a computer-based archival file for macromolecular structures. *Arch. Biochem. Biophys.*, **185**, 584–591.

DDBJ/EMBL/GenBank. (1997) *The DDBJ/EMBL/GenBank Feature Table Definition Version 2.1*.

Eilbeck,K., Brass,A., Paton,N. and Hodgman,C. (1999) INTERACT: an object oriented protein-protein interaction database. *Ismb*, **7**, 87–94.

Gasteiger,J. (1996) Chemical Information in 3D-Space. *J. Chem. Inf. Comput. Sci.*, **36**, 1030–1037.

Higgins,D.G., Thompson,J.D. and Gibson,T.J. (1996) Using CLUSTAL for multiple sequence alignments. *Methods Enzymol.*, **266**, 383–402.

Hogue,C.W., Ohkawa,H. and Bryant,S.H. (1996) A dynamic look at structures: WWW-Entrez and the molecular modeling database. *Trends. Biochem. Sci.*, **21**, 226–229.

Igarashi,T. and Kaminuma,T. (1997) Development of a cell signaling networks database. *Pac. Symp. Biocomput.*, 187–197.

Kans,J.A. and Ouellette,B.F. (1998) Submitting DNA sequences to the databases. In Baxevanis,A.D. and Ouellette,B.F. (eds), *Bioinformatics*. John Wiley & Sons, Toronto, pp. 319–353.

Marcotte,E.M., Pellegrini,M., Ng,H.L., Rice,D.W., Yeates,T.O. and Eisenberg,D. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751–753.

Mendelsohn,A.R. and Brent,R. (1999) Protein interaction methods–toward an endgame. *Science*, **284**, 1948–1950.

Mohr,E., Horn,F., Janody,F., Sanchez,C., Pillet,V., Bellon,B., Roder,L. and Jacq,B. (1998) FlyNets and GIF-DB, two internet databases for molecular interactions in *Drosophila melanogaster*. *Nucleic Acids Res.*, **26**, 89–93.

Object Management Group. (1996) *CORBA Architecture and Specifications*. OMG Publications.

Ostell,J. and Kans,J.A. (1998) The NCBI data model. In Baxevanis,A.D. and Ouellette,B.F. (eds), *Bioinformatics*. John Wiley & Sons, pp. 121–144.

Schuler,G.D., Epstein,J.A., Ohkawa,H. and Kans,J.A. (1996) Entrez: molecular biology database and retrieval system. *Methods Enzymol.*, **266**, 141–162.

Stoesser,G., Tuli,M.A., Lopez,R. and Sterk,P. (1999) The EMBL nucleotide sequence database. *Nucleic Acids Res.*, **27**, 18–24.

Sugawara,H., Miyazaki,S., Gojobori,T. and Tateno,Y. (1999) DNA Data Bank of Japan dealing with large-scale data submission. *Nucleic Acids Res.*, **27**, 25–28.

Weininger,D. (1988) SMILES, a chemical language and information system. *J. Chem. Inf. Comput. Sci.*, **28**, 31–36.