PIR-International
Protein Sequence Database (PSD)
Database Definition Document:

The Protein Sequence Component

Version CO2_6.4, November 3, 1994

David G. George

National Biomedical Research Foundation
Georgetown University Medical Center
3900 Reservoir Road, NW
Washington, DC 20007
202-687-2121
FAX: 202-687-1662
E-mail: pirmail@nbrf.georgetown.edu

Contents

Tables

1 Introduction

This document contains the database definition for the PIR-International Protein Sequence Database. The data are represented in the Sequence Database Definition Language (SDDL) [1]. The SDDL is a high-level data-definition language tailored specifically to the properties of macromolecular sequence data. The representation scheme employed in the SDDL is a superset of the CODATA sequence data exchange format [2-3]. This document assumes a basic understanding of this formalism on the part of the reader; refer to reference 1 for details. A complete SDDL database declaration is given in Appendix A; an entity--relationship diagram is provided in Appendix B. If not stated otherwise, the examples in this document have been adapted from Release 41.0, June 1994 of the PIR-International Protein Sequence Databases.

The Protein Sequence Database contains information concerning all naturally occurring *wild-type* proteins whose primary structure (the sequence) is known. In addition to sequence data, the database contains information concerning: (1) the name and classification of the protein and the organism in which it naturally occurs; (2) references to the primary literature, including information concerning the sequence determination; (3) the function and general characteristics of the protein, including gene expression, posttranslational processing, and activation; and (4) sites and regions of biological interest within the sequence. Entries in the database are cross-referenced to other related databases, including the GenBank® Genetic Sequence Data Bank [4-5], the EMBL Data Library Nucleotide Sequence Database [6], the DNA Data Base of Japan (DDBJ) [7], the GDB™ Human Genome Data Base at the Johns Hopkin's Welch Medical Library [8], the FlyBase *Drosophila* genome database [9], the LISTA yeast chromosome database [10], the Protein Data Bank at the Brookhaven National Laboratory [11], and MedLine Abstracts of the National Library of Medicine [12].

The information in the database is extracted from the published literature and supplemented with data directly submitted by research scientists. The data are *interpreted data* as defined in the National Science Foundation's Scientific Database Management report [13]. Information concerning the same protein is compared, analyzed, and merged into a single entry reflecting the most current understanding of the information. A single canonical sequence, constructed from the various reported forms, is represented in each entry. Reference-specific information specifies the conditions of the sequence determination and instructions for regenerating the reported sequence from the canonical form. All other information in the entry refers to the canonical sequence. Data concerning proteins from different species are not merged. However, information concerning different isoforms of the protein and from proteins from different strains and/or encoded by different genes may be merged when the sequences are highly similar.

Entries in the database are *dynamic*; they may be significantly modified (including both the canonical sequence and information concerning its associated properties) to reflect new understandings of the data. These changes involve separating previously merged entries (for example, when new information indicates that the sequences were isolated from different species previously thought to be separate strains of the same species), merging previously unmerged entries, and removing data shown to be incorrect and/or of dubious validity (which may involve deleting entire entries).

Much of the ancillary information concerning the proteins is stored in separate database components. These components serve as repositories for the authoritative version of information that may be shared among a number of entries, e.g., citations or formal names of species. Entries in the protein sequence component are linked to these auxiliary components and some of the information in the entries is extracted or derived from that represented in the auxiliary components. This known redundancy exists in order to present a summary of the pertinent information directly within the protein sequence entries. The corresponding information in the protein sequence component is derived from and/or checked against that in the auxiliary components by computer program. Table 1-1 lists the components referred to in this document and gives a brief description of each.

Table 1-1 Component structure of the database

Source Sequence	Sequences as reported in the primary literature
Citation	Bibliographic citations to the primary literature including submitted bodies of work and other unpublished sources
Taxonomy	Species taxonomy for the biological source
Gene Mapping	Genetic information cross-linking to the genome mapping databases
Protein Sequence	Primary component containing *canonical* protein sequence and related information

This document describes the protein sequence component.

I. General Description of Sequence Component

An example of a sequence entry is shown in Figure 1-1. An entry can be divided into several record sections. The first record section in the entry contains a single record, the ENTRY Record. This record gives a unique identification code for the entry and describes the type of sequence depicted. This section is followed by the Header Record Section, the Reference Record Section, the General Property Record Section, the Feature Record Section, and the Sequence Record Section. The Header Record Section gives general descriptive information including the protein name, biological source, and tracking information, such as creation and modification dates. The Reference Record Section contains citations to the primary literature and descriptive information pertaining specifically to each report (including the sequence as reported). The General Property Record Section gives information concerning genetics, function, and any other characteristics that are not explicitly linked to sites or regions within the sequence. The Feature Record Section gives information explicitly linked to sites or regions within the sequence. In additional to the sequence itself, the Sequence Record Section gives summary information such as sequence length, molecular weight (of the unmodified form), and checksum.

```
                    ---- Entry Record Section ----
{
  { ENTRY                 XNHUSP                          #type complete }
                    ---- Header Record Section ----
  { TITLE                 serine--pyruvate transaminase (EC 2.6.1.51), peroxisomal
                             - human }
  { ALTERNATE_NAMES       serine--pyruvate aminotransferase, peroxisomal }
  { CONTAINS              alanine--glyoxylate aminotransferase (EC 2.6.1.44) }
  { ORGANISM              #formal_name Homo sapiens #common_name man }
  { DATE                  30-Sep-1991 #sequence_revision 30-Sep-1991 ... }
  { ACCESSIONS            S10557; A38764; S14002 }
                    ---- Reference Record Section ----
  { REFERENCE             S10557
    #authors              Takada, Y.; Kaneko, N.; Esumi, H.; Purdue, P.E.;
                          Danpure, C.J.
    #journal              Biochem. J. ##volume 268 ##pages 517-520 ##year 1990
      ##title             Human peroxisomal L-alanine: glyoxylate
                             aminotransferase. Evolutionary loss of a mitochondrial
                             targeting signal by point mutation of the initiation
                             codon.
    #cross-references     MUID:90303236
    #accession            S10557 ##molecule_type mRNA  ##residues 1-392
                             ##label TAK1 ##cross-references EMBL:X53414
    #accession            A38764 ##molecule_type protein ##residues 52-61;318-330
                             ##label TAK2 }
  ...
                    ---- General Property Record Section ----
  { GENETICS              #gene GDB:SPAT }
  { COMPLEX               homodimer }
  { FUNCTION              #description aminotransferase #pathway glycine
                             biosynthesis }
  { CLASSIFICATION #superfamily serine--pyruvate aminotransferase }

                    ---- Feature Record Section ----
  { BINDING_SITE  209     #residues Lys  #bond_class covalent #ligand pyridoxal
                             phosphate #status predicted #label BS1 }

                    ---- Sequence Record Section ----
  { SUMMARY               #length 392 #molecular_weight 43010 #checksum 1797 }
  { SEQUENCE          5       10       15       20       25
              1 M A S H K L L V T P P K A L L K P L S I P N Q L L
              ... }
}
```

Figure 1-1 Sample sequence component entry

The entry was adapted from PIR-International Protein Sequence Database, Release 41.00, June 1994. Ellipses (...) indicate information omitted for display purposes.

II. Conventions Employed

In the following chapters the general characteristics of records and subrecords are described in boxed figures. These descriptions are taken from the SDDL Declaration for the protein sequence component given in Appendix A. Also listed in the boxes are constraints on the records, subrecords, and fields, including:

- *required* or *optional*
- *repeating* or *nonrepeating*
- *single valued* or *multiple valued* — for fields only

These constraints are derived from the cardinalities given in the ER diagram in Appendix B. There are several instances that are not covered by these constraints.

SDDL permits the presence of optional fields within records and subrecords. In a number of cases, all fields defined for a record or subrecord are optional, for example the ORGANISM Record and the Introns Subrecord of the GENETICS Record. The SDDL places the constraint that each record or subrecord must contain at least one data value. Hence, as a general rule, although all fields (or subfields) may be listed as optional in the boxed figure at least one must be present and contain a data value.

Similarly, the contraints imposed by the generalization hierarchies expressed in the ER model cannot be represented fully in the boxed figures. In these cases, the entity hierarchy implies that one and only one instance from a set of subentities may be present. In the boxed figures these entities are all listed as optional. Specifically, a Journal, Book, Submission, or Citation Subrecord must be present in each REFERENCE Record. No contraints are listed for these subrecords in the boxed figure on page 4-1. The constraints listed in the figures on pages 4-3 through 4-5 refer to the fields within the subrecords rather than to the subrecords themselves.

Finally, all canonical sequences must be composed from at least one source sequence. This places the contraint that in each entry there must be at least one REFERENCE Record that contains at least one Accession Subrecord containing a direct link (accession number) to the source component. This constraint is captured in the ER diagram but cannot be represented in the boxed figure notation.

2 Entry Record Section

The Entry Record Section contains a single record, the ENTRY Record.

ENTRY Record		**required, nonrepeating**
{ ENTRY	***_EntryIDY***	required, nonrepeating, single valued
#type	***_Text_Enum[]***	optional, nonrepeating, single valued
}		

The first field of the ENTRY Record contains the entry identification code, which serves as a unique identifier of the entry within the protein sequence component. These codes may change as entries are combined, decombined, or reclassified; within any given database release, however, they are unique.

The **type** field contains an enumerated value that indicates whether the sequence shown in the entry is complete or not. Table 2-1 lists the valid field values.

Table 2-1 Entry types

complete	sequence is complete as shown
fragment	sequence is incomplete

One and only one ENTRY Record is permitted per entry; the entry identification code is required and may not be repeated. The **type** field is optional, may contain only one type, and may not be repeated.

3 Header Record Section

The header records give general descriptive information including the protein name, biological source, and tracking information, such as creation and modification dates. They include the TITLE, ALTERNATE_NAMES, CONTAINS, ORGANISM, DATE, and ACCESSIONS Records

I. TITLE Record

The TITLE Record contains the title of the entry, which consists of a protein and an organism portion separated by the character string ' - '. The spaces flanking the hyphen are required. The protein and organism portions of the TITLE Record are given as brief, free-text descriptions.

TITLE Record		**required, nonrepeating**
{ TITLE }	***_Description***	required, nonrepeating, single valued

The TITLE Record is required and may not be repeated. The record must contain one and only one title.

II. ALTERNATE_NAMES Record

The ALTERNATE_NAMES Record contains alternate names for the protein.

ALTERNATE_NAMES Record		**optional, nonrepeating**
{ ALTERNATE_NAMES **}**	***_Clause_Set***	required, nonrepeating, multiple valued

The ALTERNATE_NAMES Record is optional and may not be repeated. When present, the record may contain a set of one or more alternate names.

III. CONTAINS Record

The CONTAINS Record lists the names of activities that are present within the sequence shown but are not associated with the entire sequence.

CONTAINS Record		**optional, nonrepeating**
{ CONTAINS **}**	***_Clause_Set***	required, nonrepeating, multiple valued

The CONTAINS Record is optional and may not be repeated. When present, the record may contain a set of one or more activity names.

IV. ORGANISM Record

The ORGANISM Record describes the organism in which the protein naturally occurs, i.e., the organism in which the protein is genetically expressed.

ORGANISM Record		**required, nonrepeating**
{ ORGANISM	***_EntryREF(TAX)***	optional, nonrepeating, single valued
#formal_name	***_Clause_List_Enum[]***	optional, nonrepeating, multiple valued
#common_name	***_Clause_List_Enum[]***	optional, nonrepeating, multiple valued
#variety	***_Clause_Set_Enum[]***	optional, nonrepeating, multiple valued
#note **}**	***_Clause_Set***	optional, repeating, multiple valued

The organism is also referred to as the biological source of the protein; in this usage, it is the natural source of the protein rather than the experimental source that is described. All information on the ORGANISM Record, except that in the **note** field, is standardized. The record is linked to the organism taxonomy component. This component gives a full taxonomy for the organism. The first field contains the taxonomy number; it is the entry identification code of the corresponding entry in the taxonomy component. The ORGANISM Record is required and may not be repeated; the taxonomy number is optional but no more than one may be present.

The **formal_name** field contains the formal name of the organism. Generally, the formal name is the scientific species name. For viruses, the species naming conventions differ and the formal names may not be given in Latin as is typical for the scientific names of other organisms. When the species has not been precisely determined the field may contain an ambiguous designation, such as *Aeromonas sp.* (indicating an organism of genus *Aeromonas* the species of which is unspecified), or may contain the name of a higher taxon such as *Chroococcales gen. sp.* (indicating an unspecified cyanobacterium of class *Chroococcales*). All ambiguous designations are given as special entries in the taxonomy component. In the extreme case that the organism cannot be identified, the corresponding taxonomy entry coincides with the root of the taxonomy. The **formal_name** field is followed by the **common_name** field, which contains the common name of the organism. Both fields are optional and may not be repeated; at least one of them must be present, however. These fields may contain more than one synonymous name for the same organism.

The **variety** field contains infrasubspecific subdivision names, such as strain names, isolate names, etc. This field is used only when the sequences of the same protein from different strains, etc., are significantly different and are, therefore, represented in separate entries or when the species has not been precisely determined, in which case the variety field serves to clarify the designation. The field is optional and may not be repeated. It may contain more than one variety name; these names are not synonyms. More than one variety name may be given when the sequence from more than one variety is represented in the same entry.

The **note** field may be used to provide information concerning an organism whose identity cannot be precisely determined from the information in the **formal_name**, **common_name**, or **variety** fields. This field is optional, may be repeated, and is multivalued.

V. DATE Record

The DATE Record lists the dates on which the entry was created and most recently revised.

DATE Record		**optional, nonrepeating**
{ DATE	***_Date***	optional, nonrepeating, single valued
#sequence_revision	***_Date***	optional, nonrepeating, single valued
#text_change	***_Date***	optional, nonrepeating, single valued
}		

The first field on the record is the creation date. The **sequence_revision** field gives the date of the last modification to the canonical sequence. The **text_change** field gives the date of the last modification of the text information (all information other than the sequence) in the entry. The DATE Record is optional but may not be repeated. All

date fields are optional, may contain a single date only, and may not be repeated. If the record is given, at least one date must be present.

VI. ACCESSIONS Record

The ACCESSIONS Record lists the accession numbers associated with the entry.

ACCESSIONS Record		**optional, nonrepeating**
{ ACCESSIONS **}**	***_EntryREF(NUL)_Set***	required, nonrepeating, multiple valued

This record contains a list of all accession numbers that have ever been associated with the entry. Because the policies concerning accession numbers have changed over the years, it is not possible to provide a consistent description of these data elements that would cover all cases. They are used as generalized look-up keys to help in the location of an entry in future releases of the database.

4 Reference Record Section

The Reference Record Section contains citations to the primary scientific literature and information specifically pertaining to each scientific report cited. This information includes descriptions of the contents of the report, of the conditions of the sequence determination, of the sequence as presented in the report, of detected variations in the sequence, of sequence discrepancies detected within the body of the report, and of sequence ambiguities. References are included in sequence database entries when they correspond to reports of original sequence determinations and/or reports pertaining to the assignment and/or clarification of the sequence properties, characteristics, and sites or regions of biological interest. The Reference Record Section consists of a set of one or more REFERENCE Records. At least one REFERENCE Record is required per entry but the record may be repeated.

I. REFERENCE Record

The REFERENCE Record consists of a general reference section and zero or more Accession Subrecords, which contain sequence specific information.

REFERENCE Record		**required, repeating**
{ REFERENCE	***_EntryREF(CIT)***	required,nonrepeating,single valued
#authors	***_Author_List***	required,nonrepeating,multiple valued
#journal	**...**	
#book	**...**	
#submission	**...**	
#citation	**...**	
#cross-references	***_PathREF(NUL)_Set***	optional,nonrepeating,multiple valued
#contents	***_Clause_Set***	optional,nonrepeating,multiple valued
#note	***_Clause_Set***	optional,repeating,multiple valued
#accession	**...**	
}		

The first field of the REFERENCE Record contains the reference number. The authoritative version of the citation is stored in the citation component. The reference number is the entry identification code of the citation in the citation component. Reference numbers are unique across the entire database. The record may contain one and only one reference number.

Following the reference number is the **authors** field, which contains the list of authors associated with the citation. The **authors** field is required and may not be repeated. The citation itself immediately follows. The citation is specified in either the Journal, Book, Submission, or Citation Subrecord. Only one of these fields may be specified and one is required. The Journal Subrecord contains citations to publications in scientific journals. The Book Subrecord contains citations to books or articles published in books. The Submission Subrecord contains citations to bodies of work submitted to one of the macromolecular sequence database centers. The Citation Subrecord contains citations to scientific reports that do not fit within any of the above categories.

Following the citation is the **cross-references** field. This field is optional, may not be repeated, and is multivalued; it contains a set of cross-references to other databases by citation. It is typically used to cross-reference to abstracting service databases such as MedLine. As a special case, cross-references to the Brookhaven National Laboratory's Protein Data Bank (PDB) [11] are included in this field. In the Protein Sequence Database, sets of coordinates deposited in the PDB are considered to be scientific reports and PDB coordinate set entries are directly cited. Cross-references are specified by a mnemonic identifying the database followed by an entry identification code corresponding to the entry in the specified database. The mnemonic is separated from the entry identification code by a colon. In Figure 1-1, for example, 90303236 is the Medline Unique Identifier (MUID) corresponding to the reference given in the entry. Table 4-1 gives a list of currently recognized database mnemonics.

Table 4-1 Database symbols for citation cross-references

PDB	Brookhaven National Laboratory's Protein Data Bank [11]
MUID	National Library of Medicine's MedLine Abstracts [12]

Following the **cross-references** field is the **contents** field. This field is optional, may not be repeated, and is multivalued. For papers not reporting original sequence determinations, it contains a brief description of the contents of the report, i.e., the rationale for including the report in the sequence entry. This field will serve also as a temporary repository for reference-specific information that has not been reformulated in accordance with the conventions described in this document.

The REFERENCE Record may contain an optional **note** field. In general, note fields are used to include information that cannot be formulated within the other more specific fields that have been defined for the record. Multiple **note** fields may be used or more than one note may be represented in the same field; the field is optional.

Following the **note** field are the Accession Subrecords. The Accession Subrecord is optional and may be repeated; it contains information concerning the sequence as reported in the primary literature. A separate Accession Subrecord is given for each sequence reported in the manuscript.

II. Citation Subrecords

The Journal Subrecord contains citations to articles published in scientific journals; its field value gives the recognized journal abbreviation for the corresponding journal; one and only one journal name is permitted per field.

{ REFERENCE	**...**	
#journal	***_Text_Enum[]***	required, nonrepeating, single valued
##volume	***_Ordinal***	optional, nonrepeating, single valued
##issue	***_Ordinal***	optional, nonrepeating, single valued
##pages	***_OrdinalPair***	required, nonrepeating, single valued
##year	***_Year***	required, nonrepeating, single valued
##title	***_Description***	optional, nonrepeating, single valued
...		
}		

The Journal Subrecord contains the **volume**, **issue**, **pages**, **year**, and **title** subfields, which contain the volume number, issue, pages, year, and title of the cited article, respectively. **Volume** and **issue** are optional; usually only one is included. The issue number is generally given only when there is no volume number and/or when the page numbers do not extend across issues of the same volume and therefore are not unique within a single volume. The **title** subfield is optional. These subfields may not be repeated and are single valued.

The Book Subrecord contains citations to books and to articles in books; its field value generally corresponds to the title of the book. The book citation may not always be fully parsed; in these cases, the field contains a free-text version of the citation. The field value of the book field is required, may not be repeated, and is single valued.

{ REFERENCE	**...**	
#book	***_Description***	required, nonrepeating, single valued
##editors	***_Author_List***	optional, nonrepeating, multiple valued
##pages	***_OrdinalPair***	optional, nonrepeating, single valued
##publisher	***_Text_Enum[]***	optional, nonrepeating, single valued
##address	***_Text***	optional, nonrepeating, single valued
##year	***_Year***	optional, nonrepeating, single valued
##title	***_Description***	optional, nonrepeating, single valued
...		
}		

The Book Subrecord contains the **editors**, **pages**, **publisher**, **address**, **year**, and **title** subfields. All subfields are optional and none may be repeated. With the exception of the **editors** field, all fields are single valued. The **editors** subfield contains a list of editors for the book. This field is used only for articles in books. For citations to entire books, the editors are given in the **authors** field of the REFERENCE Record. The

pages subfield is used only for articles in books, in which cases it gives the page numbers of the article. The **publisher** and **address** subfields give the name of the publishing company and the city(s) in which the book was published, respectively. The **year** subfield lists the year that the book was published. The **title** field gives the article title; it is used only for articles and is optional in these cases.

The Submission Subrecord contains citations to submitted bodies of work; its field value is the name of the database center (as listed in Table 4-2) associated with the data submission; it is required, may not be repeated, and is single valued.

{ REFERENCE	**...**	
#submission	***_Text_Enum[]***	required, nonrepeating, single valued
##month	***_Month***	required, nonrepeating, single valued
##year	***_Year***	required, nonrepeating, single valued
##description	***_Description***	optional, nonrepeating, single valued
...		
}		

When the data are submitted to a nucleic acid sequence database center, the name of the data set from which the data were extracted is generally given; only one database center name may be given and one is required.

Table 4-2 Database symbols for data submissions

Protein Sequence Database
GenBank
EMBL Data Library
DDBJ
PDB

The Submission Subrecord contains the subfields **month**, **year**, and **description**. The **month** and **year** subfields give the month and year of submission and are required. The **description** field is optional; it contains a brief description of the data submitted. None of the subfields may be repeated and all are single valued.

The Citation Subrecord contains a free-text citation when the citation does not fit into any of the above categories; the citation is required, may not be repeated, and the field is single valued. The **title** subfield gives the title for certain types of citations where titles are appropriate, i.e., Ph.D. theses. This subfield is optional, single valued, and may not be repeated.

{ REFERENCE	**...**	
#citation	***_Text***	required, nonrepeating, single valued
##title	***_Description***	optional, nonrepeating, single valued
...		
}		

III. Accession Subrecord

The Accession Subrecord of the REFERENCE Record contains information concerning the report of an original sequence determination. As more than one sequence determination may be reported within a single scientific report, the **accession** field may be repeated; it is present only when an original sequence determination is reported.

{ REFERENCE	**...**	
#accession	***_EntryREF(SRC)***	optional, nonrepeating, single valued
##status	***_Clause_Set_Enum[]***	optional, nonrepeating, multiple valued
##molecule_type	***_Clause_Set_Enum[]***	required, nonrepeating, multiple valued
##residues	***_SegmentSpec***	optional, nonrepeating, single valued
##label	***_RecordIDY***	optional, nonrepeating, single valued
##cross-references	***_PathREF(NUL)_Set***	optional, nonrepeating, multiple valued
##experimental_source	***_Text***	optional, nonrepeating, single valued
##genetics	***_RecordREF***	optional, nonrepeating, single valued
##note	***_Clause_Set***	optional, repeating, multiple valued
...		
}		

The Accession Subrecord contains the accession number; the field value is optional, may not be repeated in the same Accession Subrecord, and is single valued. All sequences processed by the Protein Sequence Database staff are stored in the source sequence component exactly as reported in the scientific literature. The accession number is the entry identification code of the corresponding sequence in the source sequence component. These numbers are unique among all reported sequences. When the sequence is not directly processed by the database staff but its determination is reported, the accession number is null. This may happen, for example, when the sequence is not explicitly shown in the report, only fragmentary sequence data are reported, or the sequence data have been submitted to another database and are not available for direct processing by the Protein Sequence Database staff. In these cases, the **status** subfield

will contain one of the values *sequence not shown*, *fragmentary data*, or *data not processed* indicating the condition.

The Accession Subrecord contains the **status**, **molecule_type**, **residues**, **label**, **cross-references**, **experimental_source**, **genetics,** and **note** subfields. Only the **note** subfield may be repeated. The **molecule_type** subfield is required. The **residues** subfield is required unless the **status** subfield contains one of the values *sequence not shown*, *fragmentary data*, *data not processed*, or *significant sequence differences*. The last status value is used to indicate that the reported sequence is significantly different from other reports of the same sequence, in which case the residues specification becomes overly complex and is not given explicitly. In this case, the accession number is given and the reported sequence data are available in the source sequence component. The **label** subfield is required whenever the **residues** subfield is present. All other subfields are optional.

The **status** field gives information concerning 1) the availability of the reported sequence data, 2) level of consistency of the data with other reports of the same sequence, 3) data verification steps employed, and 4) the level of staff review of the information. Statuses are given as short text descriptions. Table 4-3 lists the valid statuses. More than one type of status may occur in the status field separated by semicolons. Only one type from each of the four classes is permitted.

Table 4-3 Accession statuses

--- 1. Data availability ---
sequence not shown
translation not shown
fragmentary data
data not processed
--- 2 Data consistency ---
significant sequence differences
--- 3 Data verification ---
not compared with conceptual translation
compared with conceptual translation
--- 4 Review status ---
preliminary

The data availability and consistency classes have been described above. As a data verification measure, the conceptual translation of a protein sequence from the corresponding nucleic acid sequence coding region is compared with that reported in the primary literature when it is available. The data verification **statuses** indicate the completion of this step; they may not be used in conjunction with reported sequences of **molecule_type** *protein*. The review statuses indicate the level of review to which the information has been subjected. When the information is initially entered into the database it is assigned the **status** *preliminary*. After the report has been examined by senior scientific staff, this **status** is removed.

The **molecule_type** field gives the molecule type of the sequence that was experimentally determined. This field is enumerated; Table 4-4 gives a list of accepted values. In general, *mRNA* is used rather than copy DNA (cDNA) unless it can be determined that the source material was *genomic RNA*. *Nucleic acid* is used only when there is not sufficient information available to distinguish between RNA and DNA as the

molecular source and *RNA* is used only when there is not sufficient information available to distinguish between genomic RNA and mRNA.

Table 4-4 Molecule types

nucleic acid
DNA
RNA
genomic RNA
mRNA
protein

The **residues** subfield contains the residues specification, an instruction to reconstruct the reported sequence from the canonical sequence shown in the entry. The field value is of SDDL data type **_SegmentSpec**. This same data type is used for the location specifications on the sequence element feature records (refer to Section 6.I. for a description of this data type). The reconstructed sequence is identical with that stored in the source sequence component under the associated accession number. This sequence corresponds to our best interpretation of the sequence data presented in the associated scientific report. When there are discrepancies among the data presented in the report, this sequence may not correspond to that explicitly represented in any figure or table.

The **label** subfield contains a label that identifies the **accession** field; when used in conjunction with the entry identification code, this **label** may be used to uniquely identify the reported sequence within the sequence component. For example, the first residues specification shown in Figure 1-1 can be directly addressed using the *path:*

XNHUSP->TAK1

The **label** may change from release to release but it is unique within the entry.

The **cross-references** subfield contains cross-references to other macromolecular sequence databases by sequence. Cross-references are specified by a mnemonic identifying the database followed by an entry identifier. The mnemonic is separated from the entry identifier by a colon. Table 4-5 lists the currently recognized database mnemonics.

Table 4-5 Database symbols for sequence-specific cross-references

GB	GenBank Genetic Sequence Database [4]
EMBL	European Molecular Biology Laboratory Data Library [6]
DDBJ	DNA Data Base of Japan [7]
NCBIN	National Center for Biotechnology Information nucleic acid sequences [5]
NCBIP	National Center for Biotechnology Information protein sequences [5]

For the nucleic acid sequence databases (GenBank, EMBL Data Library, and DDBJ), the cross-reference is to the entry(s) containing the reported nucleic acid sequence(s) that contains the corresponding protein coding region. The nucleic acid *accession number* is given as the entry identifier. The sequences of the source nucleic acid sequence (NCBIN) and the author's conceptual translation (NCBIP) as stored in the

NCBI data sequence set extracted from scientific journals are cited by GenInfo sequence identification number.

The **experimental_source** field gives a free-text description of the experimental source of the sequence data; this field contains information concerning the details of the experimental determination. It may include the names of strains, plasmids, transposons, retrotransposons, cell-lines, varieties (or cultivars), tissues, or genomes. No distinction is made between the experimental and the natural sources of the protein at this level.

The **genetics** field is used to link to the GENETICS Record when there are multiple GENETICS Records in the entry. The field value is identical with the **label** (the first field value) of the corresponding GENETICS Record. It is used, for example, to link the reported sequence to gene expression information when sequences from different genes are represented in the same entry.

The **note** field is used to express information that cannot be formulated within other, more specific fields.

5 General Property Record Section

The general property records give information concerning function, gene expression, and any other characteristics that are not explicitly linked to sites or regions within the sequence. These records include the COMMENT, GENETICS, COMPLEX, FUNCTION, CLASSIFICATION, and KEYWORDS Records. All of the general properties records are optional.

I. COMMENT Record

The COMMENT Record contains a free-text comment concerning the canonical sequence. This record is used only when the information contained on it cannot be conveniently expressed within more specific records. The COMMENT Record may be repeated.

COMMENT Record		**optional, repeating**
{ COMMENT	***_Text***	required, nonrepeating, single valued
#label	***_RecordIDY***	optional, nonrepeating, single valued
#link	***_RecordREF_Set***	optional, nonrepeating, multiple valued
}		

The COMMENT Record may be associated specifically with other records in the entry via the **label** and **link** fields. These fields are optional and nonrepeating. The **label** field is single valued and the **link** field is multiple valued. The **label** field contains a local identifier (it must be unique within the entry) that can be used to cite the COMMENT Record from another record. For example, the **link** field of a features record may be used to associate a specific comment with a feature. The **link** field of the COMMENT Record may be used to associate the comment with one or more other records. For example, the **link** field of a COMMENT Record may be associated with several features records by listing the corresponding local identifiers (the value in the **label** field of the features record) in the **link** field of the COMMENT Record.

II. GENETICS Record

The GENETICS Record contains information relating to the gene expression of the protein represented in the entry. This record is optional and may be repeated. The GENETICS Record is repeated when the sequences of proteins from different genes, etc., are very similar and therefore are represented in the same entry. In these cases, separate GENETICS Records are given.

GENETICS Record		**optional, repeating**
{ GENETICS	***_RecordIDY***	optional,nonrepeating,single valued
#gene	***_PathREF(NUL)_Set***	optional,nonrepeating,multiple valued
#map_position	***_Text***	optional,nonrepeating,single valued
#genome	***_Text_Enum[]***	optional,nonrepeating,single valued
#gene_origin	***_Clause_Set_Enum[]***	optional,nonrepeating,multiple valued
#genetic_code	***_TableTransl(mRNA)***	optional,nonrepeating,single valued
#start_codon	***_InitCodon***	optional,nonrepeating,single valued
#introns	***_IntraCodon_Set***	optional,nonrepeating,multiple valued
##status	***_Clause_Set_Enum[]***	optional,nonrepeating,multiple valued
#other_products	***_PathREF(SRC)_Set***	optional,nonrepeating,multiple valued
#note	***_Clause_Set***	optional,repeating,multiple valued
}		

The first field of the record contains the record label, which distinguishes specific GENETICS Records when more than one appears in the entry. The field is optional, may not be repeated, and is single valued. A label is given only when more than one GENETICS Record appears in the entry.

The GENETICS Record also contains the **gene**, **map_position**, **genome**, **gene_origin**, **genetic_code**, **start_codon**, **introns**, **other_products**, and **note** fields. These fields are optional and with the exception of the **note** field may not be repeated. The **map_position**, **genome**, **genetic_code**, and **start_codon** fields are single valued; all others are multivalued.

The **gene** field contains a set of synonymous gene symbols (aliases designating the same gene). It is typical for genome mapping databases to use gene symbols as gene identification codes. We follow this point of view and provide cross-references to the genome mapping databases via the gene symbols. These cross-references are distinguished from other gene symbols by directly preceding the gene symbol with a database mnemonic followed by a colon. Figure 1, for example, displays a cross-reference to the Genome Data Base at Johns Hopkins. The currently recognized database symbols employed in this field are listed in Table 5-1.

Table 5-1 Database symbols for genetic cross-references

GDB	Human Genome Data Base at Johns Hopkin's Welch Medical Library [8]
FLY	*Drosophila* Genomic Database [9]
LISTA	Yeast Chromosome Map Database [10]

The **map_position** field gives the location of the gene within the genetic map. Conventions for designating map positions differ for different organisms; these will not be discussed here. For genes cross-referenced to specific gene mapping databases, the designation conforms to the practices of the corresponding gene mapping database centers. The map position includes a specification of the chromosome number, segment number, etc.

The **genome** field distinguishes genes that are encoded from satellite DNA or extrachromosomal material, i.e., mitochondrion, chloroplast, plasmid, transposon, etc. The field may be absent when the gene is encoded directly from the primary chromosomal genome.

The **gene_origin** field lists the origin(s) of the gene in cases where there is evidence suggesting gene-transfer, i.e., the gene originated and evolved in an organelle, organism, etc., different from that in which it is currently expressed.

The **genetic_code** field gives a symbol specifying the special genetic code used by the organism to translate the protein from its corresponding mRNA, if a genetic code other that the universal code is used.

The **start_codon** field gives the translation initiation codon if other than AUG. The field occurs only in entries that represent the form of the protein sequence that is initially translated (prior to posttranslational modification and processing). This information may be used to reconstruct the correct back-translation (the encoding mRNA) from the protein sequence. The mRNA form of the codon (U instead of T) is always given.

The **introns** field contains a list of intron specifications. Each intron specification consists of a sequence position followed by a position within the codon. The intron is situated immediately following the codon position specified. Multiple intron specifications are separated by semicolons. For example,

#introns 24/2; 45/3; 106/1

indicates that there are introns in the protein coding region between the second and third nucleotides of codon 24, between codons 45 and 46, and between the first and second nucleotides of codon 106. Note that the codon number is equal to the protein sequence position number.

The **introns** field contains one subfield, **status**, that gives the assignment status for the introns. Valid statuses are the same as those for the feature records **status** field as listed in Table 6-1. The **status** subfield is optional, may not be repeated within the same introns field, and is multiple valued.

When several products are encoded from the same gene by alternate exon splicing, the sequences are represented in separate entries when they differ significantly. The **other_products** field contains a list of accession numbers (references to the entry identification code of entries in the source sequence component) corresponding to other products encoded from the same gene.

The **note** field may be used to include genetic information that cannot be expressed in any other field on the GENETICS Record.

III. COMPLEX Record

The COMPLEX Record gives a description of the molecular associations among protein chains within a protein complex. For example, the fact that the chain whose sequence is represented in the entry self-associates to form a *homodimer* would be indicated in this record. The record is optional and may not be repeated.

COMPLEX Record		**optional, nonrepeating**
{ COMPLEX	***_Text***	required, nonrepeating, single valued
}		

IV. FUNCTION Record

The FUNCTION Record describes the function of the protein. It is optional and may be repeated.

FUNCTION Record		**optional, repeating**
{ FUNCTION	***_RecordIDY***	optional,nonrepeating,single valued
#description	***_Clause_Set_Enum[]***	optional,nonrepeating,multiple valued
#pathway	***_Clause_Set_Enum[]***	optional,nonrepeating,multiple valued
#note	***_Clause_Set***	optional,repeating,multiple valued
}		

The first field contains a label that identifies the FUNCTION Record; this field is used only when there is more than one FUNCTION Record in the entry. The FUNCTION Record also contains **description**, **pathway**, and **note** fields. These fields are optional but at least one must be present; they are multiple valued. The **description** and **pathway** fields may not be repeated. The **description** field gives a brief description of the molecular activity; the **pathway** field gives the name of the biochemical pathway in which the protein functions.

V. CLASSIFICATION Record

The CLASSIFICATION Record lists the names of the class or classes of proteins of which the protein is a member.

CLASSIFICATION Record		**optional, nonrepeating**
{ CLASSIFICATION		
#superfamily	***_Clause_Set_Enum[]***	optional, nonrepeating, multiple valued
#group	***_Clause_Set _Enum[]***	optional, nonrepeating, multiple valued
}		

Protein sequences may be classified by superfamily and/or may be assigned to groups based on function and other characteristics. The **superfamily** and **group** fields contain a list of the names of the superfamilies or groups, respectively, of which the protein sequence is a member. Both fields are optional, nonrepeating, and multiple valued; at least one must be present.

VI. KEYWORDS Record

The KEYWORDS Record lists a set of keywords that describe the protein sequence represented in the entry.

KEYWORDS Record		**optional, nonrepeating**
{ KEYWORDS	***_Clause_Set_Enum[]***	required, nonrepeating, multiple valued
}		

The keywords are used to describe properties of the molecule that cannot be represented in other general property or features records. The record is optional and may not be repeated; it contains a single field that is multivalued.

6 Feature Record Section

The feature records give information explicitly linked to sites or regions within the sequence. The position numbers specified on these records directly address residues within the canonical sequence. The feature records include the sequence element, bond, and site-specific records. The sequence element records specify information concerning sequence products, domains, regions, etc., that can be resolved as subsequence elements or transformed versions of the canonical sequence. The bond records depict pairs of residues linked by nonpeptide bonds. The site-specific records depict information associated with specific residues within the sequence, e.g., cleavage sites, binding sites, active sites, etc.

Feature records contain the fields **description**, **link**, **status**, **reference**, **note**, and **label**. Except for the **note** field, which may be repeated, these fields are optional and may not be repeated. The **description** and **label** fields are single valued; the others may contain multiple values. The first element on all feature records specifies the location of the feature; this element is followed by the **description** field. Various other feature-type specific fields may follow the **description**. The fields **link**, **status**, **reference**, **note**, and **label** follow, in order, at the end of the record.

feature **Record**		**optional, repeating**
{ *feature*	**...**	
#description	***_Description***	optional, nonrepeating, single valued
...		
#link	***_RecordREF_Set***	optional, nonrepeating, multiple valued
#status	***_Clause_Set_Enum[]***	optional, nonrepeating, multiple valued
#reference	***_EntryREF(CIT)_Set***	optional, nonrepeating, multiple valued
#note	***_Clause_Set***	optional, repeating, multiple valued
#label	***_RecordIDY***	optional, nonrepeating, single valued
}		

As a general convention, if the location of the feature is not known precisely, the location field may be omitted. For example,

{ BOND #bond_type disulfide **#bond_class** covalent **#status** experimental **}**

is a valid feature record. It indicates that disulfide bonds are known experimentally to be present but their exact locations are unknown. The location may also be omitted when the **status** field contains the value *absent*. The location field is optional and may not be repeated.

The **description** field contains a brief free-text description of the information presented on the feature record. It can be used as a title for that feature. For the PRODUCT and DOMAIN Records, it contains the name of the product or domain, respectively.

The **link** field is used to refer to other feature or general property records by label. For example, the rat pancreatic polypeptide precursor is cleaved to form pancreatic hormone and pancreatic icosapeptide. In the process, the carboxyl end of pancreatic hormone is amidated; pancreatic icosapeptide is unmodified. The following demonstrates how this information can be represented; the **link** field (via the label PCH) associates the modified site with the pancreatic hormone product specifically.

```
{ { ENTRY  PCRT      #type complete }
  { TITLE  pancreatic polypeptide precursor - rat }
  ...
  { PRODUCT 30-65   #description pancreatic hormone #status predicted
                        #label PCH }
  { SITE        65  #description amidated carboxyl end #class modified site
                        #residues Tyr #link PCH #status predicted }
  { PRODUCT 69-98   #description pancreatic icosapeptide #status predicted
                        #label PCI }
  ...
  }
```

The **status** field contains information specifying the reliability of the feature data. Table 6-1 enumerates the valid feature statuses.

Table 6-1 Feature statuses

experimental
predicted
absent
atypical
incomplete
partial abundance

Feature statuses fall into four groups: (1) experimental, predicted, and absent; (2) atypical; (3) incomplete; and (4) partial abundance. Only one status from each group is permitted; however, statuses from different groups may be used in any combination.

Experimental is used when there is some experimental evidence confirming part or all of the information listed in the record. *Predicted* is used for all cases where there is no definitive experimental evidence for the feature, i.e., it has been inferred by homology with other sequences, by pattern, and/or other computer prediction methods. *Absent* is used when there is definite (experimental) evidence to indicate that the feature is not present in the molecule. These three statuses are incompatible and may not be used in combination. *Atypical* is used to indicate a feature that does not conform with other features of that type, e.g., a homology domain that does not extend the entire length of its homologs. *Incomplete* is used to indicate that the information on the feature record, usually the location information, is incomplete; this status is used to indicate fragments, partial lists of modified sites, etc. The **status** field may be omitted on old features where the rationale for assignment is uncertain.

The **reference** field cites references by reference number and indicates that information relating to the assignment of the feature can be found in the cited paper. The **label** field contains a label that uniquely identifies a feature record within an entry. It is required on all PRODUCT and DOMAIN Records; otherwise, it is used only when the record is cited elsewhere. The **note** field contains information that does not fit into any of the other specialized fields.

I. Sequence Element Records

The sequence element records describe features that can be resolved as sequence elements or sets of sequence elements. They include the PRODUCT, DOMAIN, and REGION Records. The PRODUCT Record contains an instruction for transforming the canonical sequence into the sequence of a protein gene product. The DOMAIN Record depicts functional or structural domains. The REGION Record is used to depict any other sequence element(s).

sequence element **Record**		**optional, repeating**
{ PRODUCT	***_SegmentSpec***	optional, nonrepeating, single valued
...		
}		
{ DOMAIN	***_SegmentSpec***	optional, nonrepeating, single valued
...		
#class	***_Clause_Set_Enum[]***	optional, nonrepeating, multiple valued
...		
}		
{ REGION	***_SegmentSpec***	optional, nonrepeating, single valued
...		
}		

The location fields of the sequence element records are represented by the SDDL data type **_SegmentSpec.** A **_SegmentSpec** as an instruction to transform the canonical sequence into another sequence element or set of sequence elements via a series

of append operations. Sequence elements are represented as an ordered set of inclusive ranges of locations (or single locations) separated by commas. Sequence segments (enclosed by apostrophes) not present within the canonical sequence can be used in place of a location range. Distinct sequence elements are separated by semicolons. The order of these elements specifies their order in the transformed sequence element set. Multiple sequence elements are generally used to represent fragmentary data, e.g., when one or more segments necessary to build the transformed sequence of interest are absent from the canonical sequence.

For example, the following depicts two sequence elements.

1-24 , 'ASN' , 27 ; 35-64

The first sequence element is constructed by extracting the first 24 residues from the canonical sequence and appending *Ala-Ser-Asn* followed by residue 27 of the canonical sequence; the second sequence element consists of residues 35 to 64 of the canonical sequence.

Domains are regions of the sequence that can be distinguished by defined criteria. Domains are characterized by domain class type, e.g., sequence homology, structural homology, etc. The **class** field of the DOMAIN Record describes the domain class type; currently, *homology* (sequence homology) is the only recognized value for this field. The field is omitted for all other domain class types. The **class** field is optional, may not be repeated, and is multivalued.

The REGION Record is used to depict regions of the sequence that do not correspond to gene products or domains.

II. Bond Records

The BOND Record depict pairs of residues linked by nonpeptide bonds. Bonds are depicted between or within the polypeptide chains.

Bond Record		**optional, repeating**
{ BOND	***_Bond_Set***	optional, nonrepeating, multiple valued
...		
#bond_type	***_Text_Enum[]***	optional, nonrepeating, single valued
#residues	***_Residue_List***	optional, nonrepeating, multiple valued
#bond_class	***_Text_Enum[]***	optional, nonrepeating, single valued
#partner	***_BondPartner()***	optional, nonrepeating, single valued
...		
}		

BOND Records contain the fields **bond_type**, **residues**, **bond_class**, and **partner**. All of these fields are optional and may not be repeated. The **residues** field is multiple valued; the others are single valued. The location field of the BOND Record may contain a set of distinct bond specifications separated by semicolons. The field is optional, nonrepeating, and multiple valued.

BOND Records represent a chemical bond between two residues. There are three general forms: 1) bonds between residues in the same sequence; 2) bonds between

residues in different sequences; and 3) bonds between a residue of the source sequence and an undefined residue of a sequence partner.

The first type of bond is represented by a pair of locations separated by a dash. For example, the following record shows a set of bonds between residues 21 and 35, 87 and 102, and 112 and 145 of the same sequence.

{ BOND 21-35;87-102;112-145 **#bond_type** disulfide **#bond_class** covalent **#status** experimental **}**

The second type of bond is represented by a location (in the canonical sequence) and a specification of a location in a sequence from a different entry separated by a dash. Locations in different entries are specified by listing the entry identification code (optionally preceded by a component identifier followed by a colon, e.g., PIR1:FGHUB) followed by the location within the sequence enclosed in parentheses. For example, the following record describes a bond between residue 55 of the canonical sequence and residue 95 of the sequence represented in entry FGHUB.

{ BOND 55-FGHUB(95) **#bond_type** disulfide **#bond_class** covalent **#status** experimental **}**

Note that when the exact location of the bond in the second sequence in not known the second location (and the enclosing parentheses) may be omitted. For example,

{ BOND 55-FGHUB **#bond_type** disulfide **#bond_class** covalent **#status** experimental **}**

A special case of this type of bond is used to represent bonds between protein chains of the same type. For example, bovine cGMP-dependent protein kinase forms a homodimer linked by a disulfide bond between residues 42 of the two chains of the dimer. Both copies of the (identical) chain are stored in the same entry. This situation is represented by omitting the entry identification code from the specification of the second location while retaining the parentheses, i.e., specifying, by default, the entry in which the canonical sequence is represented.

{ BOND 42-(42) **#bond_type** disulfide **#bond_class** covalent **#status** experimental **}**

The third type of bond is represented by a single location. This representation is used when the residue within the partner sequence is not known with certainty or the sequence of the partner is uncertain or unavailable. The name of the partner sequence is given in the **partner** field. For example, the following record from the mouse laminin chain B2 entry depicts a disulfide bond between laminin chain B2 and laminin chain B1.

{ BOND 1598 **#bond_type** disulfide **#bond_class** covalent **#partner** laminin chain B1 **#status** experimental **}**

The **bond_type** field gives a chemical description of the bond. A partial list of valid bond types is given in Table 6-2.

Table 6-2 Bond types

(2S,3S,6R)-3-methyl-lanthionine
(S)-S-(2-aminovinyl)cysteine
5-imidazolone
cysteinylhistidine
cysteinyltyrosine
desmosine
disulfide
isopeptide
lysinoalanine
sn-(2S,6R)-lanthionine
thiolester
tryptophan-tryptophyl quinone

The **residues** field gives a list (by amino acid symbol) of the residues involved in the bond(s). These residues are listed in the same order as specified by the sites listed in the location field. There is a one-to-one correspondence between residues and location sites. Residues are separated by semicolons.

The **bond_class** field characterizes the electronic configuration of the bond. Recognized bond classes are listed in Table 6-3.

Table 6-3 Bond classes

covalent
axial ligand
hydrogen
ionic
noncovalent

The **partner** field gives a free-text description of the bonding partner if other than the canonical sequence; generally, the proper name of the partner chain is given.

III. Site-specific Records

Sites are groups of residues of biological interest specified by their sequence position. They are depicted in the site-specific records, which include the CLEAVAGE_SITE, SITE, BINDING_SITE, and ACTIVE_SITE Records. The CLEAVAGE_SITE Record depicts sites within the protein where chemical cleavage may occur. The BINDING_SITE Record describes sites where exogenous chemical elements bind to the protein. The ACTIVE_SITE Record depicts the active site of enzymes. The SITE Record is used to depict all other types of sites.

A site is specified as a single location or as a range of locations separated by a hyphen. Even when specified as a range, sites are treated as individual residues. For example,

1-3

specifies the residues at positions 1, 2, and 3. The site-specification allows for two levels of grouping as distinguished by commas or semicolons. Site locations separated by

commas are grouped and groups of site locations are separated by semicolons. For example,

1-3 , 5 ; 6 , 7

designates two groups of sites: residues 1, 2, 3, and 5 constitute the first group and residues 6 and 7 make up the second. The meaning of this grouping depends upon the record type.

All site-specific records contain the **residues** field, which explicitly lists each residue involved in the site or group of sites; this field is optional, may not be repeated, and is multivalued. There is a one-to-one correspondence between the locations listed in the site specification and the residues in the list; both lists are ordered. As a special exception, when all residues within the site are the same, only one residue must be explicitly represented.

Cleavage Site Record

Cleavage sites are sites where the protein chain may be cleaved by enzymatic or other chemical processes. These sites are depicted on the CLEAVAGE_SITE Record.

Cleavage Site Record		**optional, repeating**
{ CLEAVAGE_SITE	***_IntraSite_Set***	optional, nonrepeating, multiple valued
...		
#residues	***_Residue_List***	optional, nonrepeating, multiple valued
#agent	***_Clause_Set_Enum[]***	optional, nonrepeating, multiple valued
...		
}		

Chemical cleavage may occur at the peptide bond between amino acids residues or at a backbone bond within a residue. Hence, cleavage sites differ from other sites in that they correspond to sites between amino acid residues or to sites within a residue rather than to the residues themselves. When the cleavage occurs between amino acids, the locations of the flanking amino acids are given in the cleavage site specification separated by a dash. When the cleavage occurs at an internal amino acid bond, only the single residue location is depicted. Multiple cleavage sites may be represented on the same record separated by semicolons. For example, the following record indicates that cleavage occurs within residue 24, between residues 34 and 35, and between residues 67 and 68.

{ CLEAVAGE_SITE 24; 34-35; 67-68 **#status** experimental **}**

The **agent** field of the CLEAVAGE_SITE Record lists the chemical or biological agent involved in cleavage of the bond. This field is optional, may not be repeated, and may be multiple valued.

Site Record

The SITE Record is used to depict sites of biological interest that do not correspond to those defined by the other site-specific records.

Site Record		**optional, repeating**
{ SITE	***_SiteCluster***	optional, nonrepeating, single valued
...		
#class	***_Text_Enum[]***	optional, nonrepeating, single valued
#residues	***_Residue_List***	optional, nonrepeating, multiple valued
...		
}		

Sites are characterized by site class. The **class** field of the SITE Record is used to designate the type of site class to which the site belongs. This field is optional, nonrepeating, and single valued. Recognized site classes are listed in Table 6-4.

Table 6-4 Site classes

	inhibitory
	modified

Binding Site Record

Binding sites are sites where exogenous chemical factors bind. These sites are depicted by the BINDING_SITE Records.

Binding Site Record		**optional, repeating**
{ BINDING_SITE	***_BindingSite***	optional, nonrepeating, single valued
...		
#bond_type	***_Text_Enum[]***	optional, nonrepeating, single valued
#residues	***_Residue_List***	optional, nonrepeating, multiple valued
#bond_class	***_Text_Enum[]***	optional, nonrepeating, single valued
#ligand	***_Clause_List_Enum[]***	optional, nonrepeating, multiple valued
...		
}		

Binding sites represent chemical bonds between one or more residues and a set of exogenous chemical factors (**ligands**). Binding sites consist of clusters of site groups; each of which binds to a distinct ligand. Note that each site within a group of binding sites binds to the same ligand, not to a distinct ligand of the same type. The ligands are listed

in the **ligand** field; this field is optional, nonrepeating, and multiple valued. When present, there is a one-to-one correspondence between ligands in the ligand list and site groups within the binding site as indicated by their respective orders. In the special case where there is only one type of ligand, it is permitted to abbreviate the list and specify only a single representative explicitly. For example, the following record specifies two distinct groups of binding sites.

{ BINDING_SITE 11,14,17,58; 21,48,51,54 **#ligand** 4Fe-4S **#status** experimental **}**

The first group of sites consists of residues 11, 14, 17, and 58 and binds to a single iron--sulfur center (symbolized by 4Fe-4S); the second group consists of residues 21, 48, 51, and 54 and binds to a second iron--sulfur center of the same type.

The BINDING_SITE Record contains the **bond_type** and **bond_class** fields, which are defined as described for the BOND Record.

Active Site Record

An active site is defined here as the set of residues involved in enzymatic catalysis; typically, their electrons (or electrons from a covalently attached cofactor) participate in the catalytic reaction. Note that this definition differs from that generally employed by crystallographers; it more closely reflects the biochemistry of the enzyme.

Active Site Record		**optional, repeating**
{ ACTIVE_SITE	***_SiteCluster***	optional, nonrepeating, single valued
...		
#residues	***_Residue_List***	optional, nonrepeating, multiple valued
...		
}		

Multiple active sites of the same type may be represented on a single ACTIVE_SITE Record; the site grouping convention of the site specification is used to distinguish distinct active sites. For example, the following record designates two active sites, both involving histidine, glutamine, and aspartic acid: the first comprises residues 32, 45, and 46 and the second comprises residues 123, 148, and 149.

{ ACTIVE_SITE 32,45-46; 123,148-149 **#residues** His; Gln; Asp; His; Gln; Asp **}**

7 Sequence Record Section

The sequence records give the sequence and summary values computed from it, including the sequence length, the molecular weight of the sequence, and a checksum computed on it. The sequence records consist of the SUMMARY Record and the SEQUENCE Record. Both records are required and may not be repeated.

Summary Record		**required, nonrepeating**
{ SUMMARY		
#length	***_SeqLength***	required, nonrepeating, single valued
#molecular_weight	***_SeqMolWgt***	optional, nonrepeating, single valued
#checksum	***_SeqCheckSum***	required, nonrepeating, single valued
}		

The SUMMARY Record contains the **length**, **molecular_weight**, and **checksum** fields. The **length** field gives the length of the canonical sequence, the **molecular_weight** field gives the molecular weight of its unmodified form, i.e., the molecular weight computed from the sequence shown in the entry. This field is given only in entries of **type** *complete*. The **checksum** field gives a checksum computed on the sequence by the method introduced by Devereux, et al. [14] and that is specified by the CODATA format [2]. With the exception of the **molecular_weight** field, all fields are required. None of the fields may be repeated and all are single valued.

Sequence Record		**required, nonrepeating**
{ SEQUENCE	***_Sequence***	required, nonrepeating, single valued
}		

The SEQUENCE Record contains the canonical sequence. It is represented using the one-letter amino acid abbreviations recommended by the IUPAC-IUB Commission on Biochemical Nomenclature [15].

8 Appendix A

SDDL Declaration for the Protein Sequence Component

```
                    --- Entry Record Section ---

{
  { ENTRY                          _EntryIDY
      #type                        _Text_Enum[Entry_type]
      }

                    --- Header Record Section ---

  { TITLE                          _Description
      }

  { ALTERNATE_NAMES                _Clause_Set
      }

  { CONTAINS                       _Clause_Set
      }

  { ORGANISM                       _EntryREF(TAX)
      #formal_name                 _Clause_List_Enum[Tax_sci]
      #common_name                 _Clause_List_Enum[Tax_com]
      #variety                     _Clause_Set_Enum[Tax_var]
      #note                        _Clause_Set
      }

  { DATE                           _Date
      #sequence_revision           _Date
      #text_change                 _Date
      }

  { ACCESSIONS                     _EntryREF(NUL)_Set
      }
```

--- Reference Record Section ---

{ REFERENCE	***_EntryREF(CIT)***
#authors	***_Author_List***
#journal	***_Text_Enum[Jrn]***
##volume	***_Ordinal***
##issue	***_Ordinal***
##pages	***_OrdinalPair***
##year	***_Year***
##title	***_Description***
#book	***_Description***
##editors	***_Author_List***
##pages	***_OrdinalPair***
##publisher	***_Text_Enum[Publisher]***
##address	***_Text***
##year	***_Year***
##title	***_Description***
#submission	***_Text_Enum[DBCenters]***
##month	***_Month***
##year	***_Year***
##description	***_Description***
#citation	***_Text***
##title	***_Description***
#cross-references	***_PathREF(NUL)_Set***
#contents	***_Clause_Set***
#note	***_Clause_Set***
#accession	***_EntryREF(SRC)***
##status	***_Clause_Set_Enum[Status_acc]***
##molecule_type	***_Clause_Set_Enum[MolType]***
##residues	***_SegmentSpec***
##label	***_RecordIDY***
##cross-references	***_PathREF(NUL)_Set***
##experimental_source	***_Text***
##genetics	***_RecordREF***
##note	***_Clause_Set***
}	

--- General Property Record Section ---

{ COMMENT	***_Text***
#label	***_RecordIDY***
#link	***_RecordREF_Set***
}	
{ GENETICS	***_RecordIDY***
#gene	***_PathREF(NUL)_Set***
#map_position	***_Text***
#genome	***_Text_Enum[Genome]***
#gene_origin	***_Clause_Set_Enum[GeneOrigin]***
#genetic_code	***_TableTransl(mRNA)***
#start_codon	***_InitCodon***
#introns	***_IntraCodon_Set***
##status	***_Clause_Set_Enum[Feature_stat]***
#other_products	***_PathREF(SRC)_Set***
#note	***_Clause_Set***
}	
{ COMPLEX	***_Text***
}	
{ FUNCTION	***_RecordIDY***
#description	***_Clause_Set_Enum[Keyword_function]***
#pathway	***_Clause_Set_Enum[Keyword_pathway]***
#note	***_Clause_Set***
}	
{ CLASSIFICATION	
#superfamily	***_Clause_Set_Enum[Class_sup]***
#group	***_Clause_Set_Enum[Class_group]***
}	
{ KEYWORDS	***_Clause_Set_Enum[Keyword_keyword]***
}	

--- Feature Record Section ---

-- sequence element records --

{ PRODUCT	***_SegmentSpec***
#description	***_Description***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

{ DOMAIN	***_SegmentSpec***
#description	***_Description***
#class	***_Clause_Set_Enum[Domain_class]***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

{ REGION	***_SegmentSpec***
#description	***_Description***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

-- bond records --

{ BOND	***_Bond_Set***
#description	***_Description***
#bond_type	***_Text_Enum[Bond_type]***
#residues	***_Residue_List***
#bond_class	***_Text_Enum[Bond_class]***
#partner	***_BondPartner(Bond_prtn)***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

-- site-specific records --

{ CLEAVAGE_SITE	***_IntraSite_Set***
#description	***_Description***
#residues	***_Residue_List***
#agent	***_Clause_Set_Enum[Cleavage_agent]***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

{ SITE	***_SiteCluster***
#description	***_Description***
#class	***_Text_Enum[Site_class]***
#residues	***_Residue_List***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

{ BINDING_SITE	***_BindingSite***
#description	***_Description***
#bond_type	***_Text_Enum[Bond_type]***
#residues	***_Residue_List***
#bond_class	***_Text_Enum[Bond_class]***
#ligand	***_Clause_List_Enum[Ligands]***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

{ ACTIVE_SITE	***_SiteCluster***
#description	***_Description***
#residues	***_Residue_List***
#link	***_RecordREF_Set***
#status	***_Clause_Set_Enum[Feature_stat]***
#reference	***_EntryREF(CIT)_Set***
#note	***_Clause_Set***
#label	***_RecordIDY***
}	

--- Sequence Record Section ---

```
{ SUMMARY
   #length                          _SeqLength
   #molecular_weight                _SeqMolWgt
   #checksum                        _SeqCheckSum
   }

{ SEQUENCE                          _Sequence
   }

}
```

9 Appendix B

ER Diagram of the Protein Sequence Component

The entity-relationship (ER) diagram given in this appendix conforms to the conventions of Batini et al. [16], with several extensions as described here.

The ER model employs three types of abstractions: classification, aggregation, and generalization. Classification is used for defining one concept as a class of objects characterized by common properties. Aggregation defines a class from a set of other classes that represent components of the superclass. Generalization is used to define subset relationships among two or more classes.

Classification and aggregation are used for building data structures within databases. In the ER model, individual items of information are classified by field type. Field types are known as **attributes**, which form the basic conceptual units of the ER model. Attributes are aggregated into **entities**, which represent classes of real world objects. Entities may be further aggregated by **relationships**, which provide a mapping between two or more entities. Relationships may also have associated attributes, although this feature is not used here.

An **instance** is a dynamic, time-variant data element or collection of data elements that conforms to the structure defined by attributes, entities, and relationships. An instance of an entry identification code attribute, for example, is a specific entry identification code. Instances of attributes, entities, and relationships constitute the actual data in the database. In general, attributes, entities, and relationships are time-invariant (they change only when the database schema is modified); they describe the structure of the data.

Entities are represented in the ER diagram as rectangles; the entity name appears within the rectangle. Attributes are indicated by small open circles (**o**); the attribute names immediately follow. An entity may contain one or more special attributes or groups of attributes called **keys**, whose values serve to uniquely identify instances of the entity. Two types of keys are employed here: **primary** and **secondary keys**. Entry identification codes serve as the **primary keys** of the principal entity (the entry) of each database component. As defined in the SDDL, the *entry* is the primary grouping of information in a database component. A *component instance* consists of a set of entry instances each with the same structure and interdata relationships. The entry identification keys are **simple** (composed of a single attribute) and are **internal** (entirely contained within a single entity, the entry). Entry identification codes are always required, only one value is associated with each entry instance, and they are unique

within the component. The **primary key** is distinguished from other attributes in the diagram by using a large **X** rather than an **o** as the attribute designator.

Subentities may also contain **local identifiers**, e.g., the feature labels. These identifiers are generally optional and are unique within each entry. They are unique across the entire database component only in combination with the primary key. The **secondary keys** are **composite**: they consist of the primary key and the local identifier. These keys are **mixed** external and internal; the primary key is external to the entity (the feature) and the local identifier is contained within it. Rather than explicitly representing composite secondary keys in the following diagrams, local identifiers have been distinguished from other attributes by using a small **x** rather than an **o** as the attribute designator. It is to be understood, however, that the secondary key corresponds to the combination of the primary key and the local identifier.

Relationships are designated in the ER diagrams using small ovals. All relationships employed here are **binary** (between two entities) and are directional. Three general types of relationships are employed: (1) *Contains* relationships between entities within the same entry instance. These relationships are implicitly defined by the structure of the entry (2) *Is linked to* relationships between entities within the same entry instance, these relationships are explicitly represented via **local identifiers**. (3) *Is directly linked to* relationships between an entity and an entry in a different component**.** These relationships are explicitly defined by **primary key**. Rather than specifically labeling relationships in the following diagrams, we distinguish different relationship types using different graphic representations for the pairs of lines connecting the relationships with their associated entities. *Contains* relationships are denoted by solid lines; *is linked to* relationships are designated by light dotted lines; *is directly linked to* relationships are designated by heavy dashed lines. The source entity (i.e., the containing entity of a *contains* relationship) is depicted to the left of or on top of the target entity (i.e., the entity that is contained).

Linking entities in *is linked to* relationships contain link attributes, whose values are identical with some local identifier and serve as pointers to entity instances by label. For example, the value of the genetics attribute (in the Accession Subrecord) of the REFERENCE Record would be identical with the label attribute of the corresponding GENETICS Record. Entities are *directly linked to* entries in different components by primary key. For example, the value of the reference-number attribute of the REFERENCE Record links the reference in the sequence component entry directly to the corresponding citation component entry by the entry identification code of the reference entry in the citation component.

The **cardinality** of an entity-relationship pair is the number of times that an instance of an entity may be involved in the relationship. The minimal and maximal cardinalities of entity-relationship pairs are indicated in the diagram by two integers enclosed in parentheses and separated by a comma. **n** is used to indicate an unspecified cardinality greater than one. For example, the REFERENCE Record may or may not contain an Accession Subrecord; multiple Accession Subrecords may be contained in the same REFERENCE Record. Hence, the minimum and maximum cardinality of the REFERENCE Record with respect to the corresponding *contains* relation is **0** and **n**. On the other hand, an Accession Subrecord must be contained in at least 1 REFERENCE Record and the same Accession Subrecord may occur within 1 REFERENCE Record only; hence, the minimum and maximum cardinalities of the Accession Subrecord with respect to this relationship are **1**.

The SDDL formalism ensures that every instance of the target entity (the entity that is contained) involved in a *contains* relationship is directly linked to a specific instance of the containing entity. The target instance exists only when contained and that particular instance may be contained in only one containing instance. Thus, the minimum and maximum cardinalities of target instances with respect to these relationships are always **1**; nevertheless, the cardinalities are listed in the diagrams for completeness.

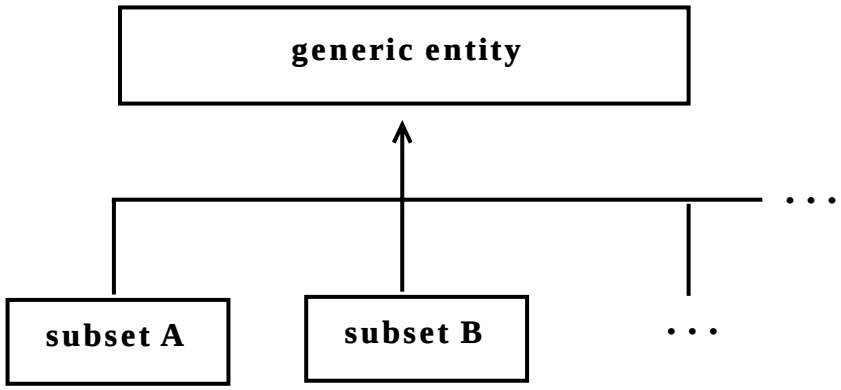
The relationship properties, many-to-one, one-to-one, etc., are inversely related to the maximum cardinalities of the entities involved in the relationship. If entity A has maximum cardinality 1 and entity B has a maximum cardinality greater than one then the relationship between A and B is said to be many-to-one. Likewise, if A has a maximum cardinality of 1 and B has a maximum cardinality of 1 then the relationship between A and B is one-to-one. For example, the *is linked to* relationship between the ORGANISM Record entity of the sequence component and the taxonomy component entity is many-to-one; there are many occurrences of the same species name on different ORGANISM Records (in different entries) in the protein sequence component but each ORGANISM Record is linked to a single entry in the taxonomy component.

The minimal cardinality specifies whether or not an entity is optional or required. If the minimum cardinality of the containing entity of a *contains* relationship is 0 then the secondary entity is optional; if it is greater than 0 then the secondary entity is required. Likewise, if the minimum cardinality of the linking entity in a *is linked to* relationship is zero than the link to the auxiliary component is optional; otherwise, it is required. For example, the REFERENCE Record Section link to the source sequence component (via the Accession Subrecord) is optional whereas its link to the citation component is required.

Minimum and maximum cardinalities are also assigned to the association of attributes with entities. In this case, if the minimum cardinality is 0, the attribute is optional; otherwise, it is required. If the maximum cardinality is 1, the attribute may not be repeated. The ER model does not define collective attribute properties, such as set and list, as are employed in the SDDL. The ER model assumes that attributes are single valued. As an extension to the formalism, a third integer has been added to the attribute cardinality pair that indicates the maximum number of values that may occur within an attribute. When this integer is greater than **1**, the attribute is multi-valued. For example, the reference number attribute of the REFERENCE Record is single valued, whereas the authors attribute is multi-valued.

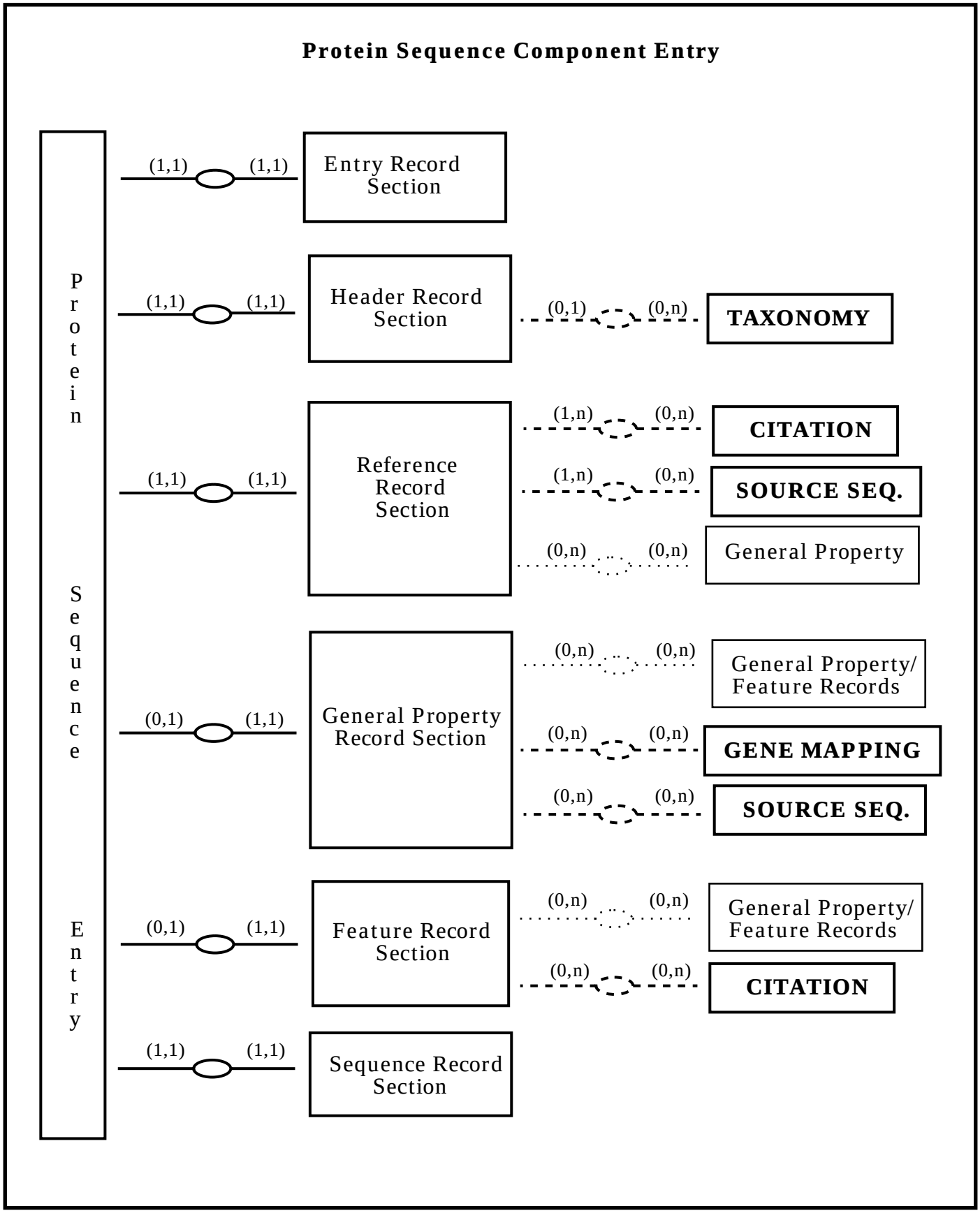
The Accession Subrecord of the REFERENCE Record provides an example of the associations implied by the SDDL formalism and their correlations with the cardinalities. The Accession Subrecord is optional and may occur more than once in a REFERENCE Record. Hence, the minimum and maximum cardinalities of the *contains* relationship of the REFERENCE Record with respect to the Accession Subrecord are **0** and **1**, respectively. When the Accession Subrecord is present, the accession number is not required, the accession number attribute may not be repeated within a subrecord and each attribute may contain at most one accession number (the accession number attribute is single valued). Hence, the cardinalities of the accession number attribute are 0**, 1**, and **1**, respectively. Note that in order to preserve the hierarchical structure of the record whenever the Accession Subrecord is present the identifier **#accession** must be presented in the REFERENCE Record, even when the accession number is absent (or null). When an accession number is present it must link to one and only one entry in the source sequence component. Hence, the cardinalities of the *is linked to* relationship of the accession number attribute with respect to the source sequence component are **1** and **1**, respectively.

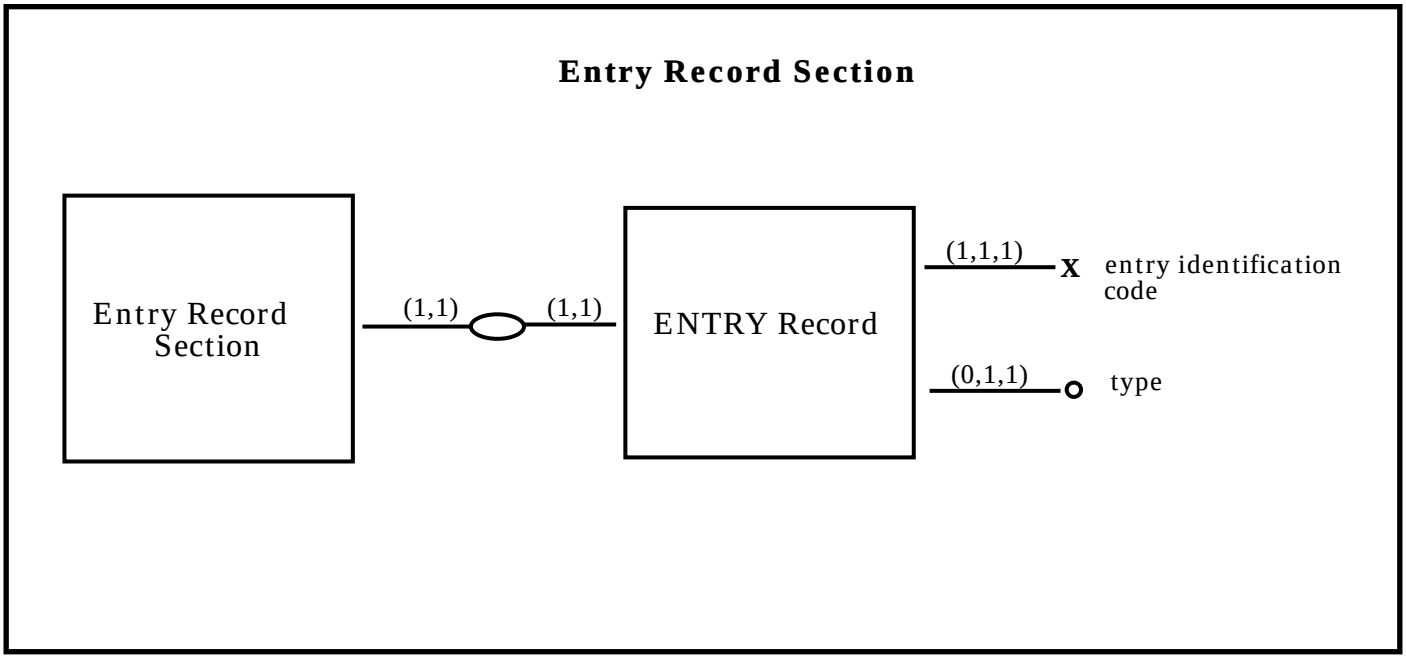
The ER model also includes the concept of generalization abstraction, which establishes a mapping from a generic class (entity) to a subset or set of subsets of that class. The citation entity is an example of a generic entity; instances of this entity may be journal, book, submission, or citation entities. Generalization hierarchies are indicated in the ER diagram using the following symbolism to connect the generic entity with its constituent subsets.

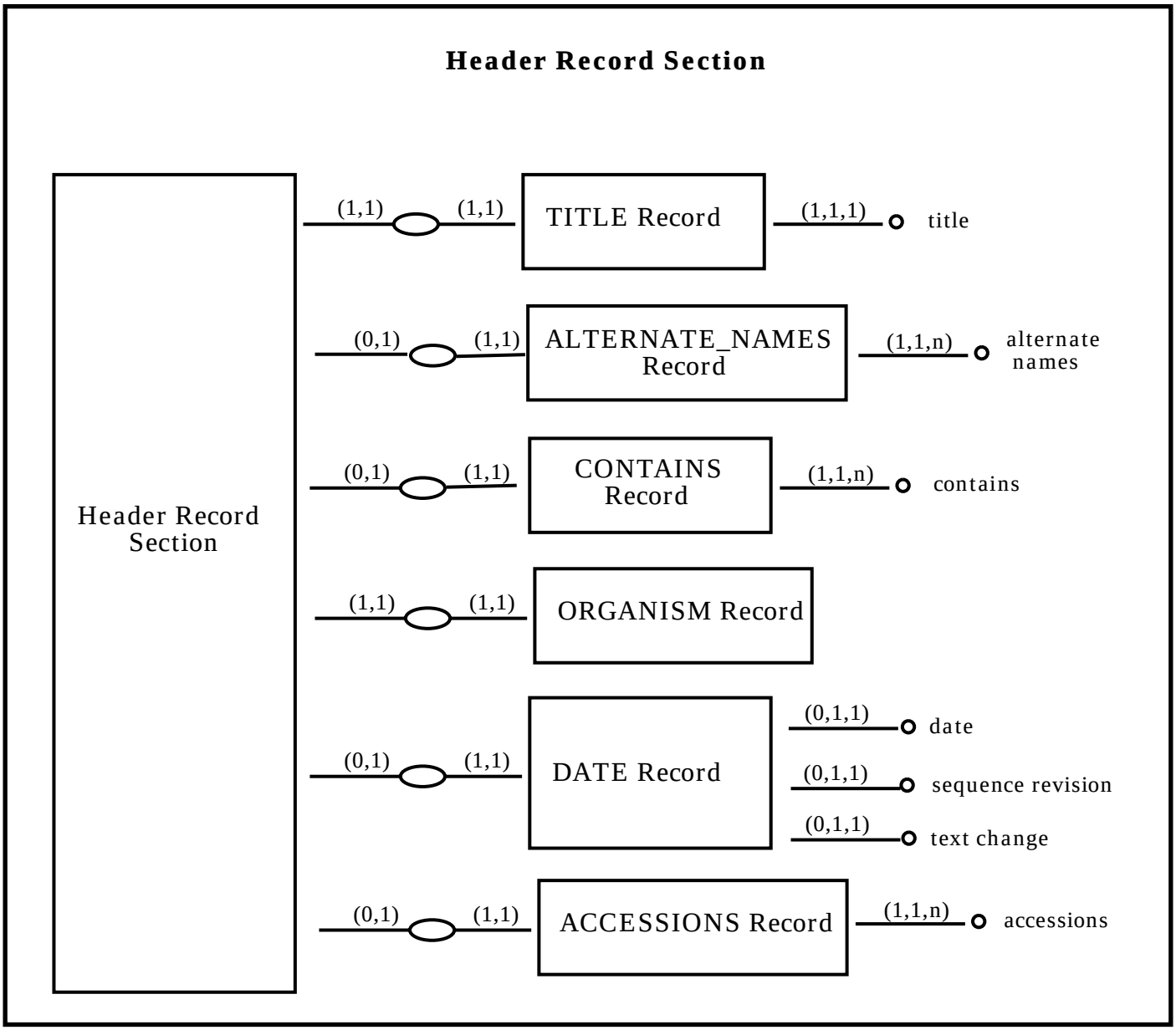


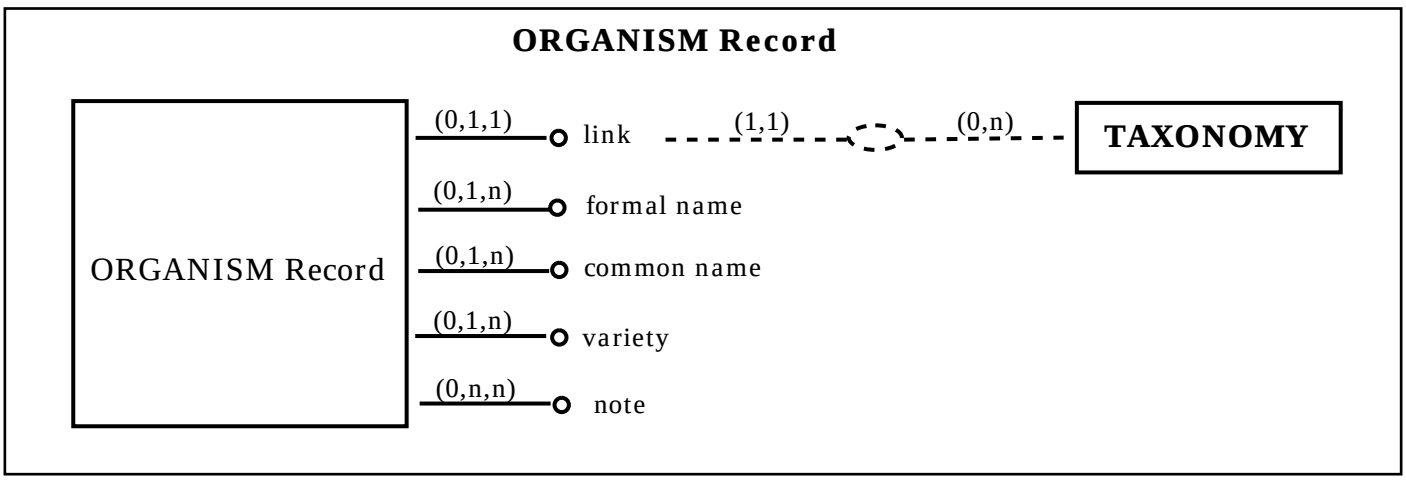
Attributes are **inherited** within generalization hierarchies. The progeny classes inherit the attributes of the parent. For example, the description, link, status, reference, note, and label attributes of a Feature Record entity are inherited by all features, i.e., they are attributes of the sequence element, bond, and site-specific records.

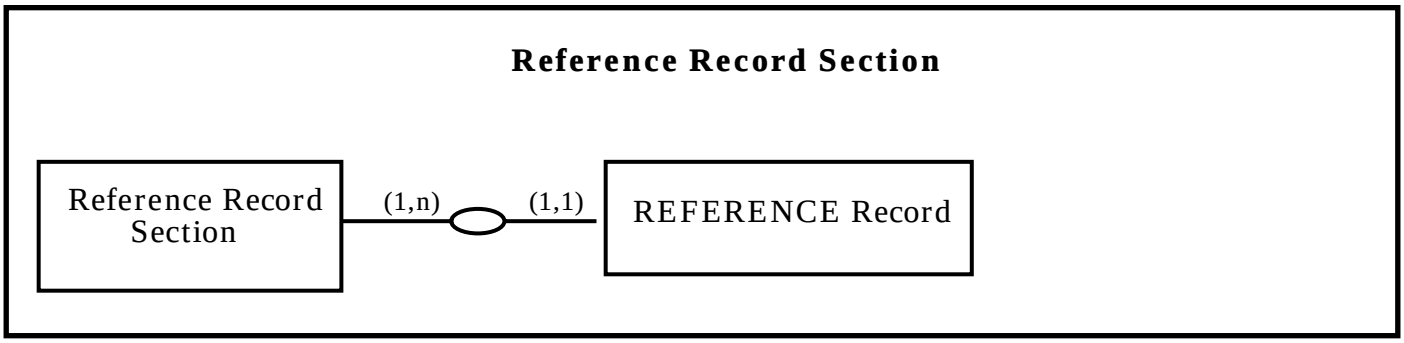
The ER model allows two types of coverage properties to be associated with generalization abstractions: **total (t)** or **partial (p)** coverage and **exclusive (e)** or **overlapping (o)** coverage. The coverage is **total** if every instance of the generic class is mapped to at least one subset class. The coverage is **exclusive** if each instance of the generic class is mapped to at most one subset class; it is **overlapping** if there exists some instance of the generic class that maps to two or more subset classes. All class coverages employed here are total and exclusive; nevertheless, the coverage type is explicitly represented in the diagram for completeness. A total and exclusive generalization corresponds to a **partitioning** of the generic class into subsets. The citation entity is partitioned into journal, book, submission, or citation entities. As the minimum and maximum cardinality of the REFERENCE Record with respect to the *contains Citation Subrecord* relationship are 1, the ER diagram indicates that the REFERENCE Record must contain one and only one instance of either a journal, book, submission, or citation entity.

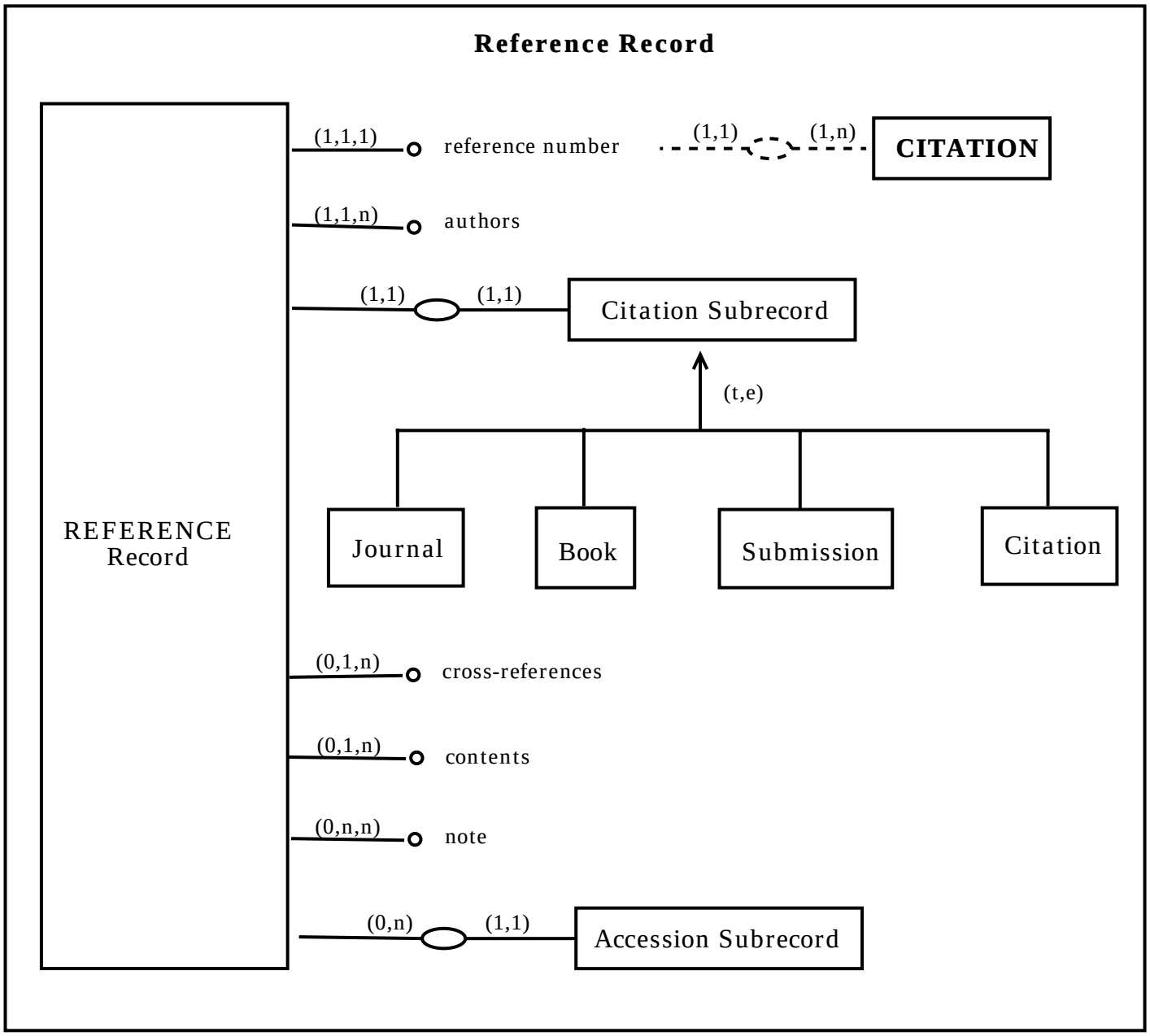

Protein Sequence Component Entry
Protein Sequence Entry
(1,1)
(1,1)
Entry Record Section
(1,1)
(1,1)
Header Record Section
(0,1)
(0,n)
TAXONOMY
(1,1)
(1,1)
Reference Record Section
(1,n)
(0,n)
CITATION
(1,n)
(0,n)
SOURCE SEQ.
(0,n)
(0,n)
General Property
(0,1)
(1,1)
General Property Record Section
(0,n)
(0,n)
General Property/ Feature Records
(0,n)
(0,n)
GENE MAPPING
(0,n)
(0,n)
SOURCE SEQ.
(0,1)
(1,1)
Feature Record Section
(0,n)
(0,n)
General Property/ Feature Records
(0,n)
(0,n)
CITATION
(1,1)
(1,1)
Sequence Record Section


Entry Record Section
Entry Record Section
(1,1)
(1,1)
ENTRY Record
(1,1,1)
X
entry identification code
(0,1,1)
type

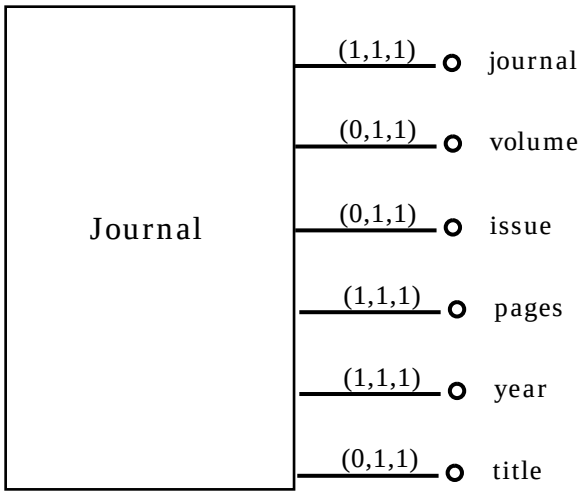

Header Record Section
Header Record Section
(1,1)
(1,1)
TITLE Record
(1,1,1)
title
(0,1)
(1,1)
ALTERNATE_NAMES Record
(1,1,n)
alternate names
(0,1)
(1,1)
CONTAINS Record
(1,1,n)
contains
(1,1)
(1,1)
ORGANISM Record
(0,1)
(1,1)
DATE Record
(0,1,1)
date
(0,1,1)
sequence revision
(0,1,1)
text change
(0,1)
(1,1)
ACCESSIONS Record
(1,1,n)
accessions


ORGANISM Record
ORGANISM Record
(0,1,1)
link
(1,1)
(0,n)
TAXONOMY
(0,1,n)
formal name
(0,1,n)
common name
(0,1,n)
variety
(0,n,n)
note

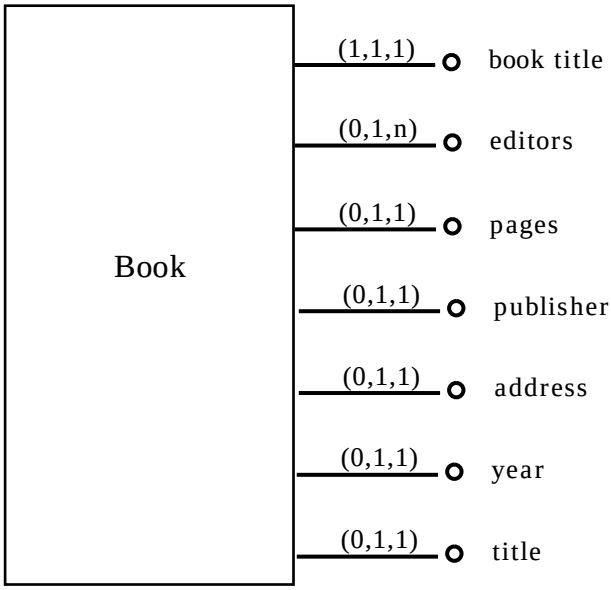

Reference Record Section
Reference Record Section
(1,n)
(1,1)
REFERENCE Record

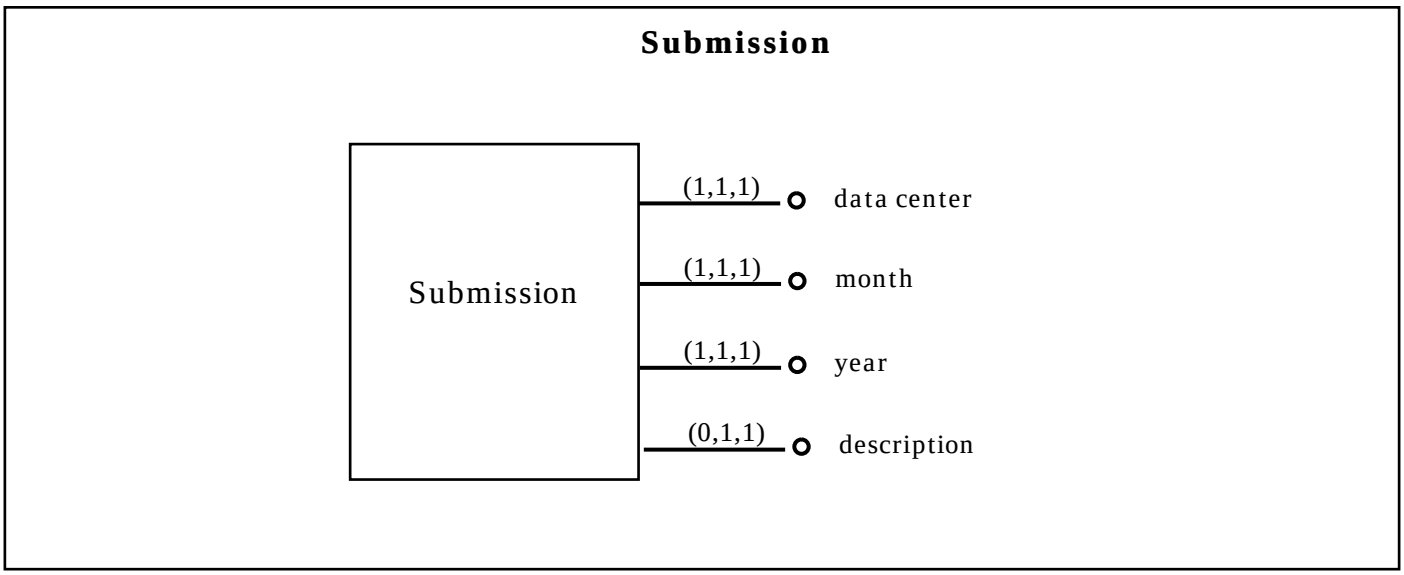

Reference Record
REFERENCE Record
(1,1,1)
reference number
(1,1)
(1,n)
CITATION
(1,1,n)
authors
(1,1)
(1,1)
Citation Subrecord
(t,e)
Journal
Book
Submission
Citation
(0,1,n)
cross-references
(0,1,n)
contents
(0,n,n)
note
(0,n)
(1,1)
Accession Subrecord

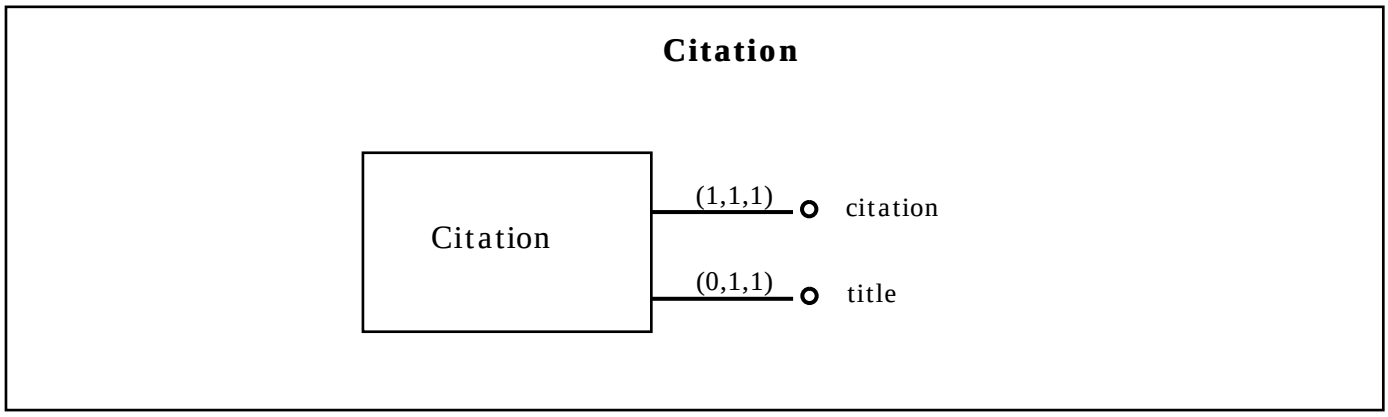
Journal

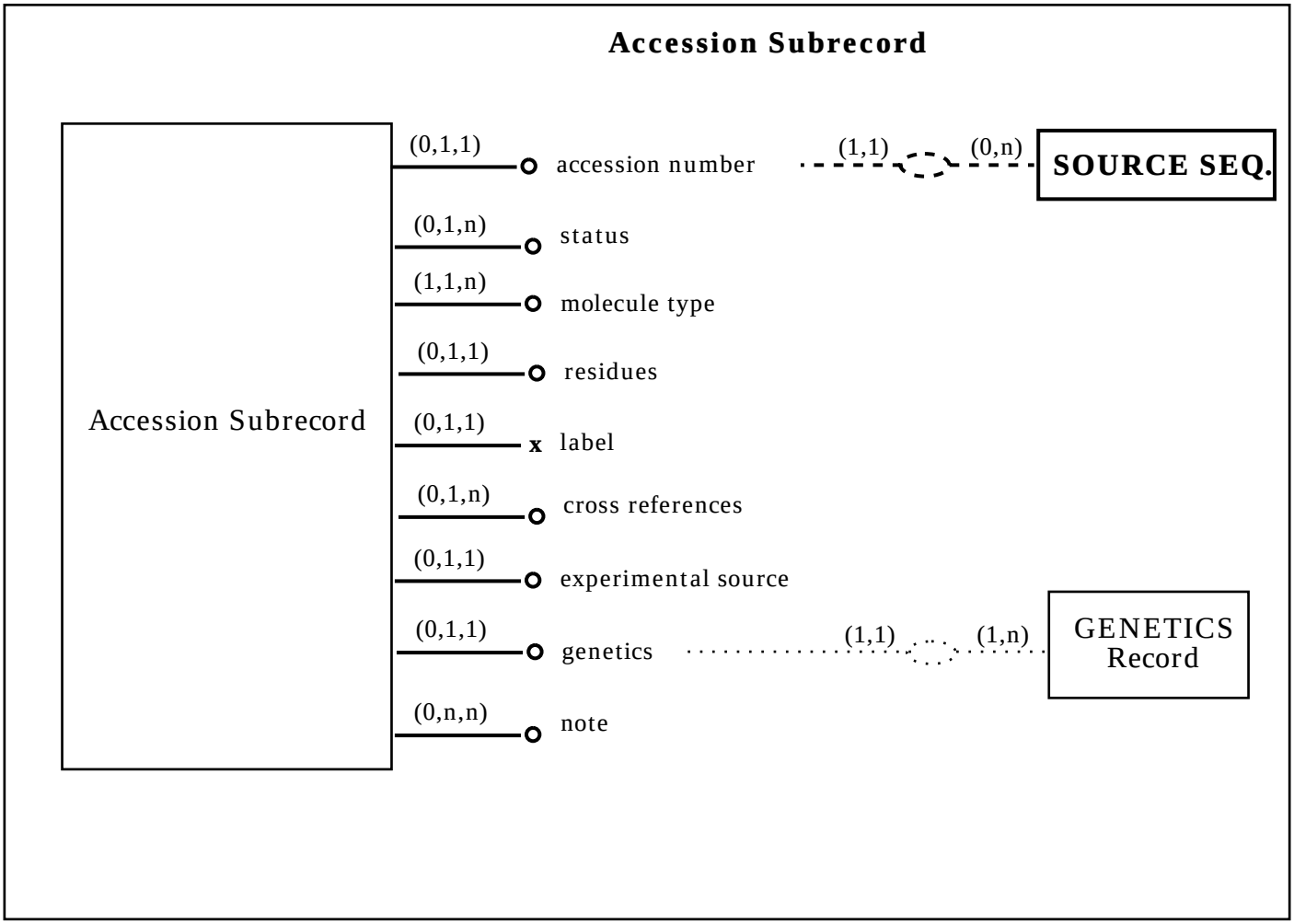


Book

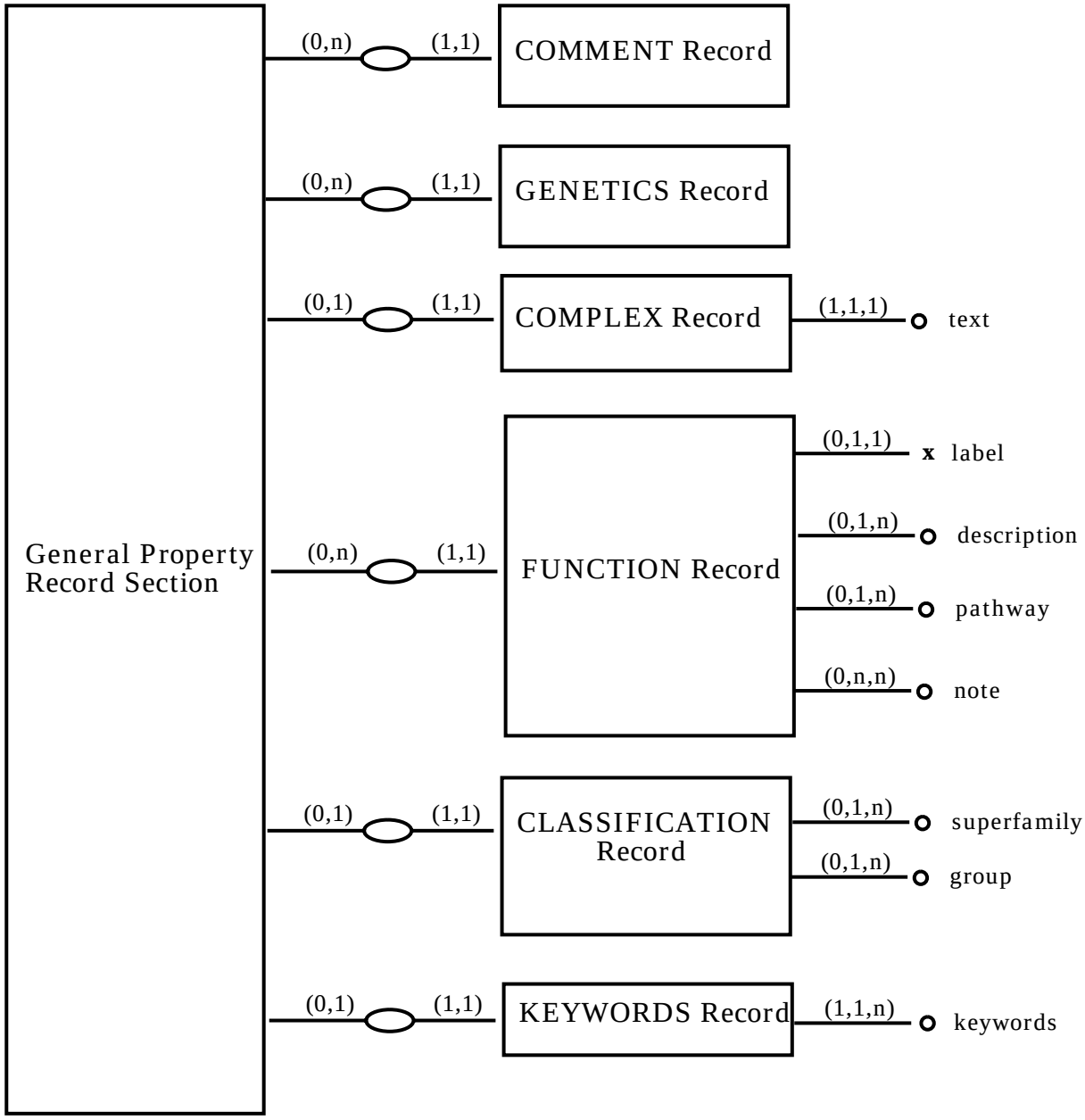


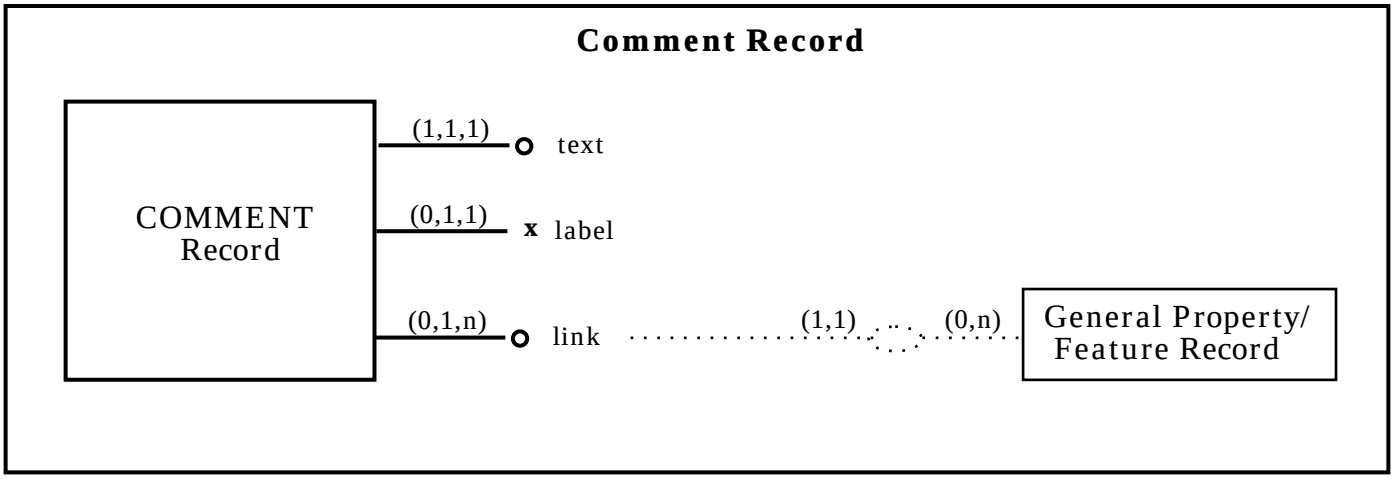

Submission
Submission
(1,1,1)
data center
(1,1,1)
month
(1,1,1)
year
(0,1,1)
description

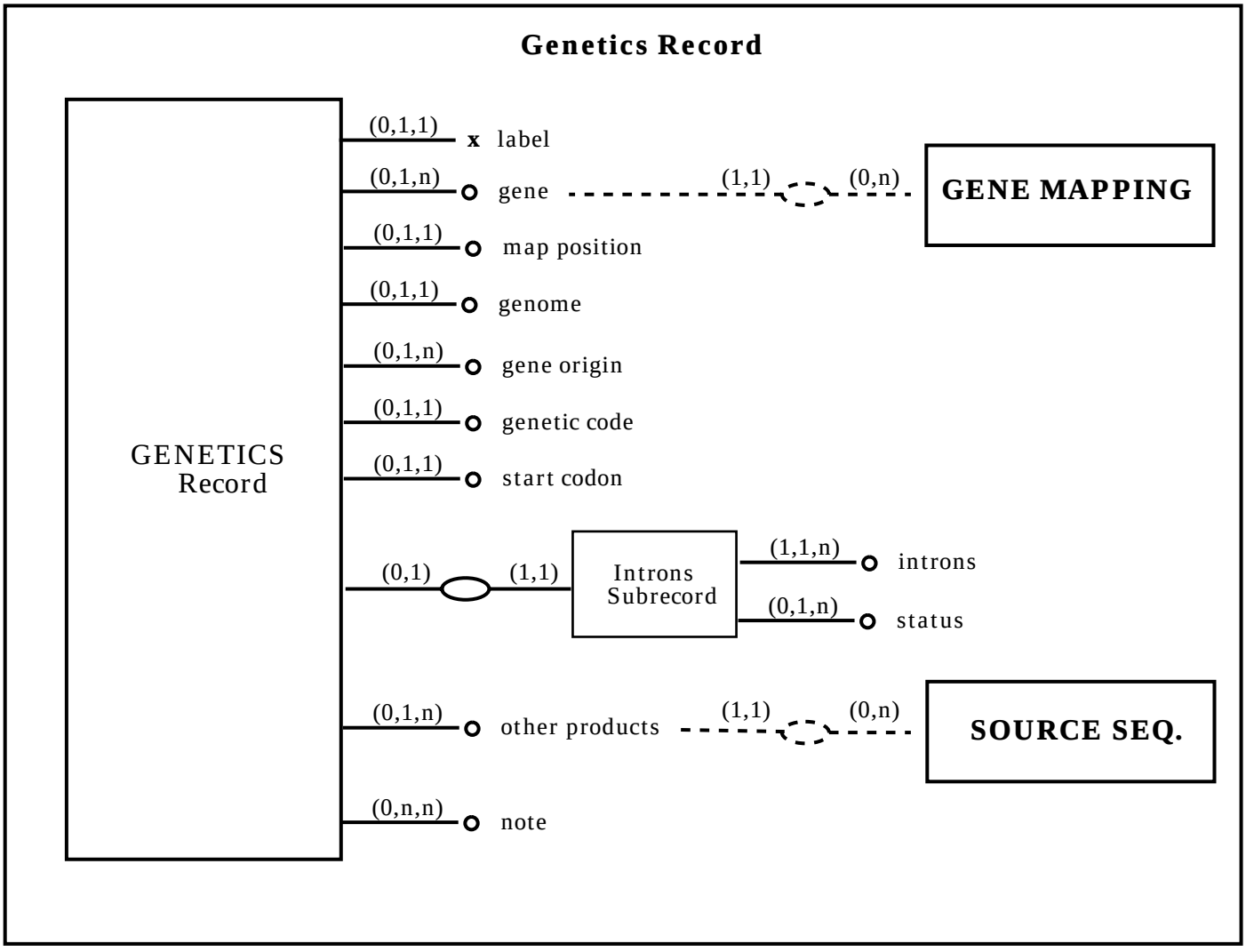

Citation
Citation
(1,1,1)
citation
(0,1,1)
title

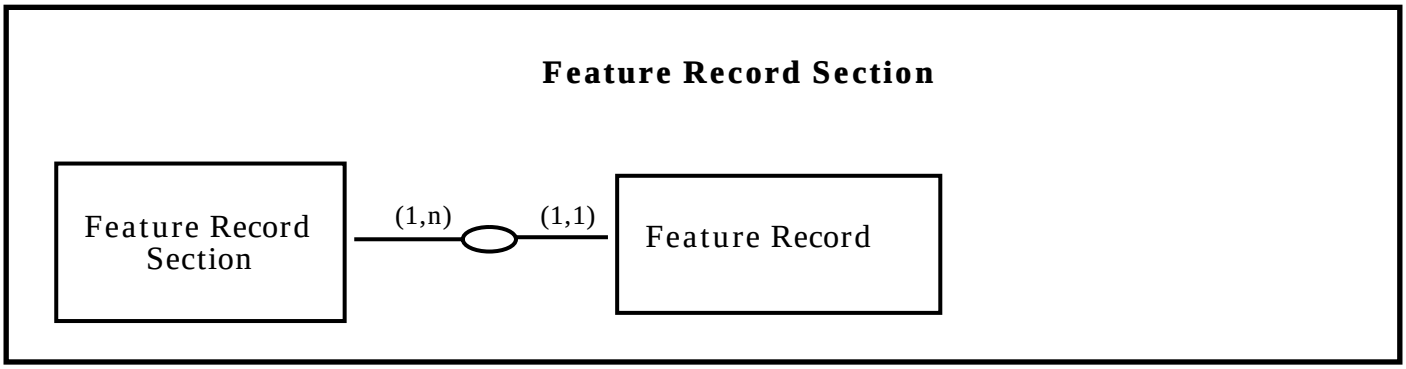

Accession Subrecord
Accession Subrecord
(0,1,1)
accession number
(1,1)
(0,n)
SOURCE SEQ.
(0,1,n)
status
(1,1,n)
molecule type
(0,1,1)
residues
(0,1,1)
x
label
(0,1,n)
cross references
(0,1,1)
experimental source
(0,1,1)
genetics
(1,1)
(1,n)
GENETICS Record
(0,n,n)
note

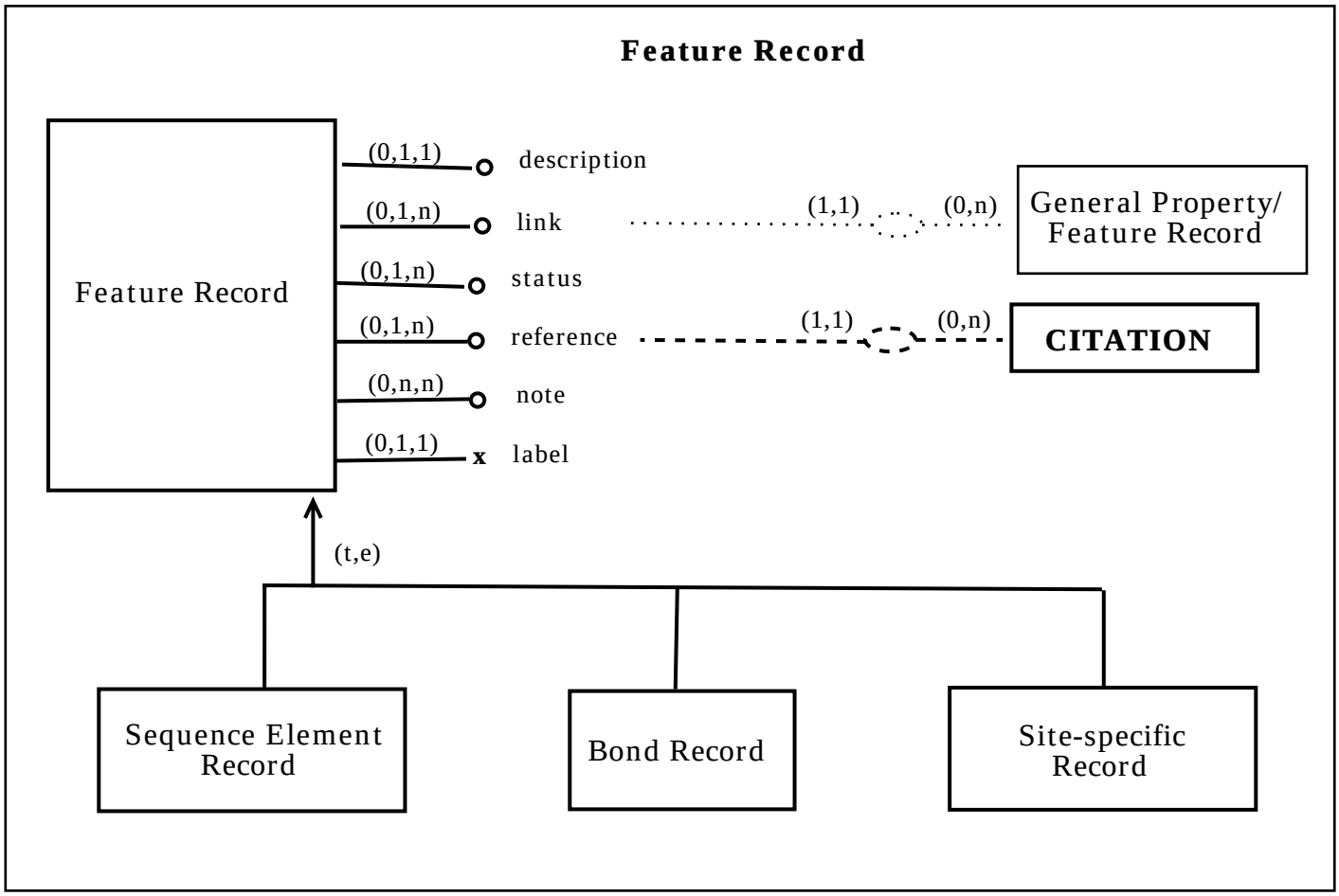
General Property Record Section

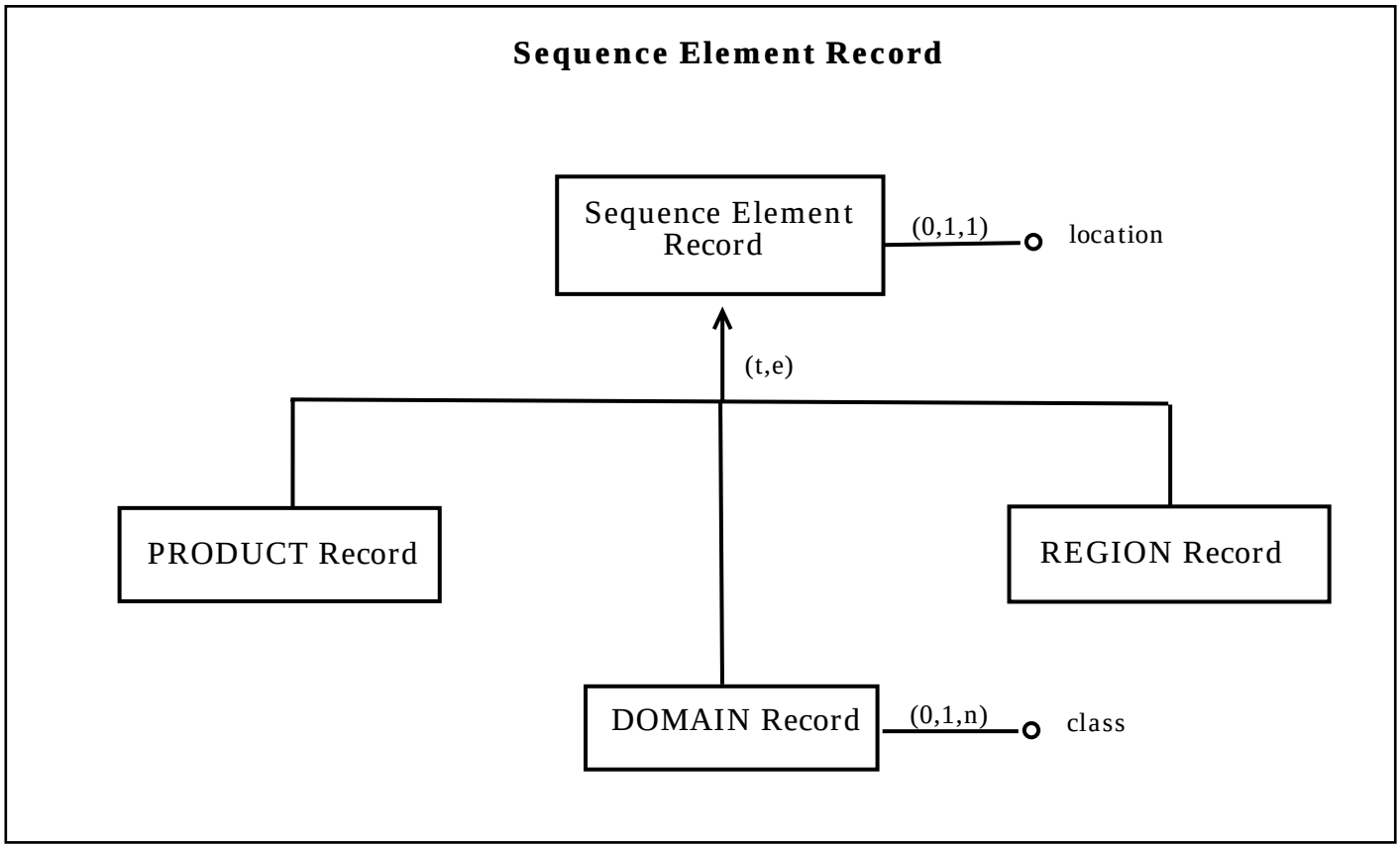


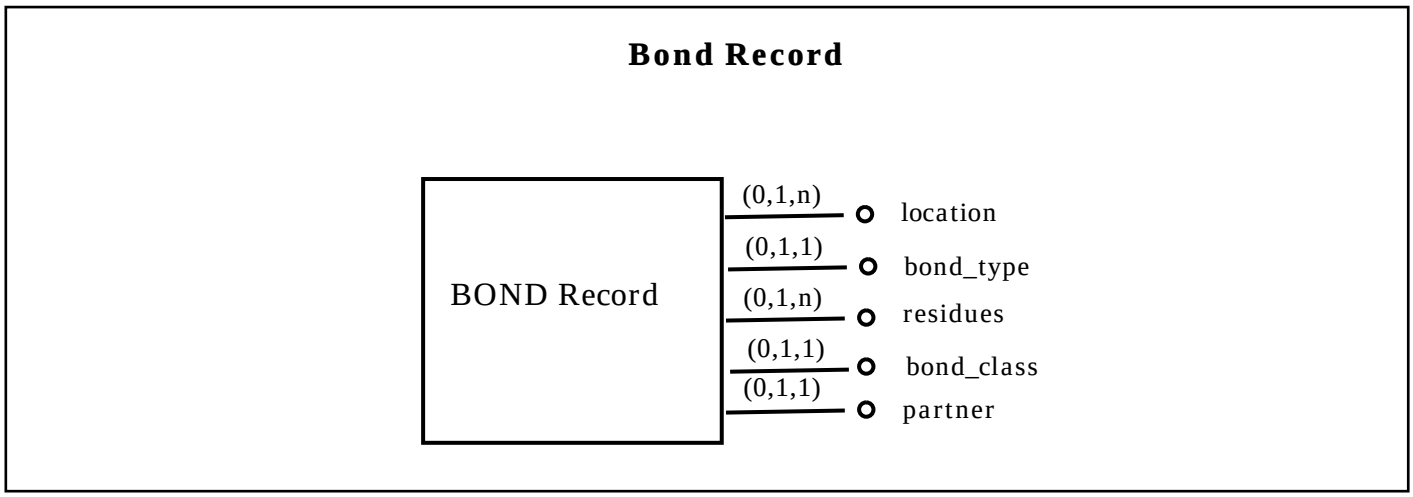

Comment Record
COMMENT Record
(1,1,1)
text
(0,1,1)
x
label
(0,1,n)
link
(1,1)
(0,n)
General Property/ Feature Record

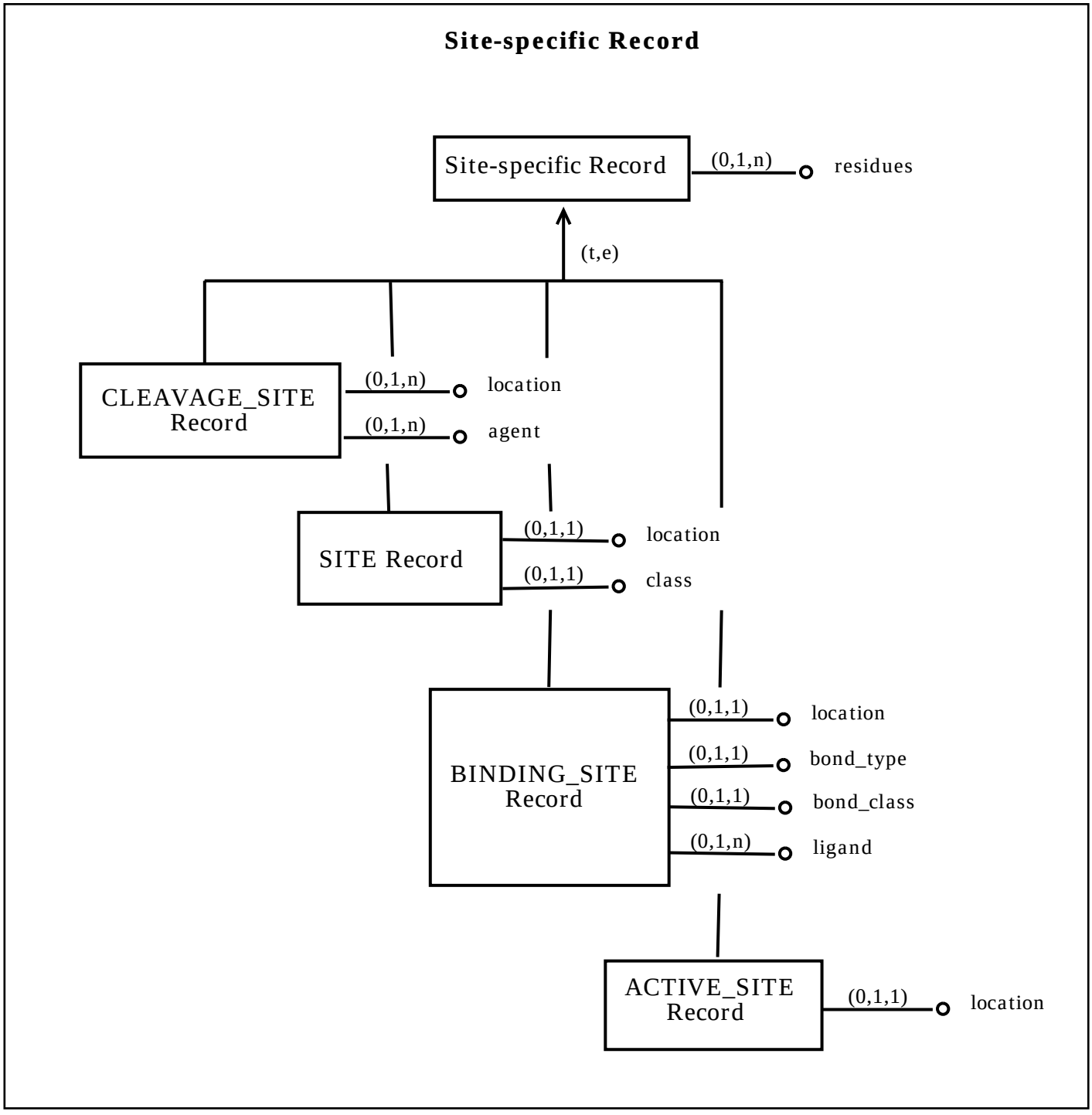

Genetics Record
GENETICS Record
(0,1,1)
x
label
(0,1,n)
gene
(1,1)
(0,n)
GENE MAPPING
(0,1,1)
map position
(0,1,1)
genome
(0,1,n)
gene origin
(0,1,1)
genetic code
(0,1,1)
start codon
(0,1)
(1,1)
Introns Subrecord
(1,1,n)
introns
(0,1,n)
status
(0,1,n)
other products
(1,1)
(0,n)
SOURCE SEQ.
(0,n,n)
note

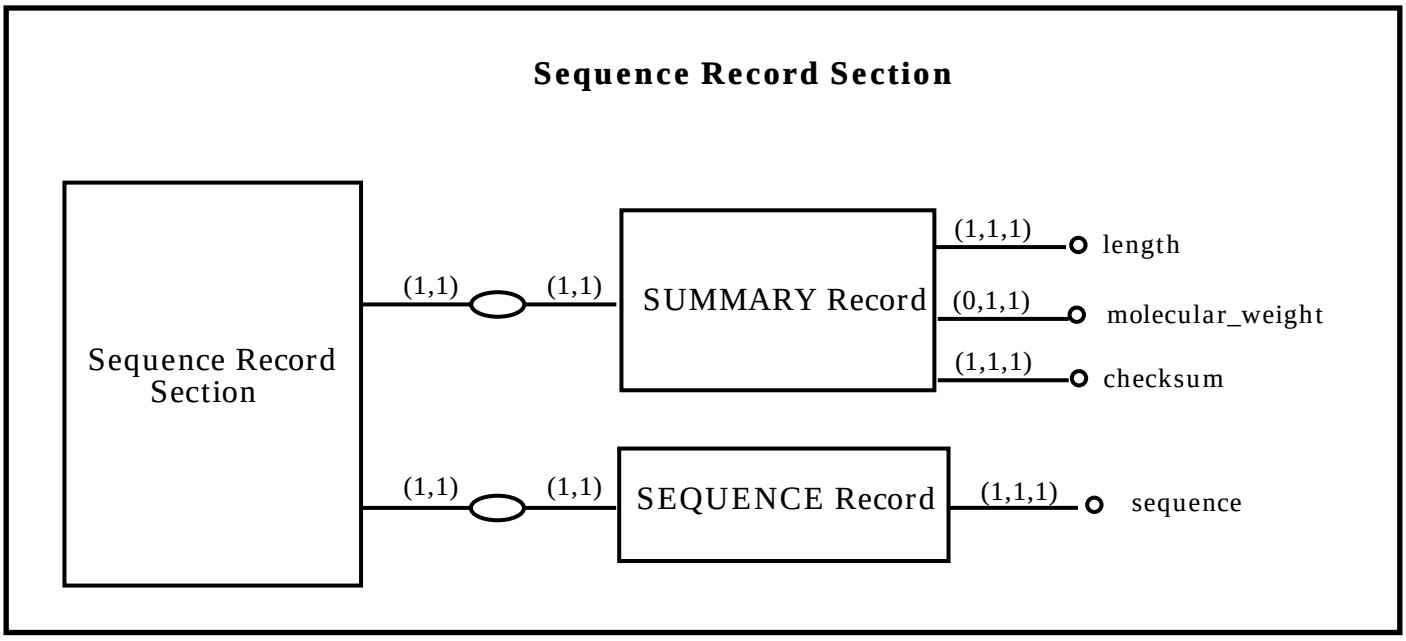

Feature Record Section
Feature Record Section
(1,n)
(1,1)
Feature Record


Feature Record
Feature Record
(0,1,1)
description
(0,1,n)
link
(1,1)
(0,n)
General Property/ Feature Record
(0,1,n)
status
(0,1,n)
reference
(1,1)
(0,n)
CITATION
(0,n,n)
note
(0,1,1)
x
label
(t,e)
Sequence Element Record
Bond Record
Site-specific Record


Sequence Element Record
Sequence Element Record
(0,1,1)
location
(t,e)
PRODUCT Record
REGION Record
DOMAIN Record
(0,1,n)
class


Bond Record
BOND Record
(0,1,n)
location
(0,1,1)
bond_type
(0,1,n)
residues
(0,1,1)
bond_class
(0,1,1)
partner


Site-specific Record
Site-specific Record
(0,1,n)
residues
(t,e)
CLEAVAGE_SITE Record
(0,1,n)
location
(0,1,n)
agent
SITE Record
(0,1,1)
location
(0,1,1)
class
BINDING_SITE Record
(0,1,1)
location
(0,1,1)
bond_type
(0,1,1)
bond_class
(0,1,n)
ligand
ACTIVE_SITE Record
(0,1,1)
location


Sequence Record Section
Sequence Record Section
(1,1)
(1,1)
SUMMARY Record
(1,1,1)
length
(0,1,1)
molecular_weight
(1,1,1)
checksum
(1,1)
(1,1)
SEQUENCE Record
(1,1,1)
sequence

10 Appendix C

Compatibility with NBRF format version 6.0

The NBRF format was developed in the late 1970's and further extended in the early 1980's [18-20]. From its origin the format was intended to support access to the data on Digital Equipment Corporation (DEC) VAX/VMS systems, although the data are represented as ASCII characters and access is not restricted to these systems. A sample entry in NBRF format is given in Figure 10-1.

The data are organized into entries that contain all the information associated with a particular sequence, including the title, the biological source, references, associated text, and the sequence itself. Each entry consists of a variable number of consecutive *physical* records. The records are of variable length up to 500 characters (to fit within one VAX/VMS block). The information contained in these lines is divided into four sections that are listed in the order that they occur in the entry.

1 Header (exactly 1 record)
Information that marks the line as the first line of an entry and that identifies the sequence contained in the entry.

2 Title (exactly 1 record)
The protein and organism name.

3 Sequence (variable number of records)
The amino acid sequence.

4 Text (variable number of records)
All other information contained in the entry.

The data may be formulated as a single file, in which the records occur exactly as given above, or as two files: one containing the text information and one containing the sequence information. In the later case, the text file contains the Header, Title, and Text Sections while the sequence file contains the Header, Title, and Sequence Sections. In all cases, the entries are listed consecutively in the same order.

```
>P1;XNHUSP
serine--pyruvate transaminase (EC 2.6.1.51), peroxisomal - human
 MASHKLLVTPPKALLKPLSIPNQLLLGPGPSNLPPRIMAAGGLQMIGSMSKDMYQIMDEIKEGIQ
^YVFQTRNPLTLVISGSGHCALEAALVNVLEPGDSFLVGANGIWGQRAVDIGERIGARVHPMTKDP
^GGHYTLQEVEEGLAQHKPVLLFLTHGESSTGVLQPLDGFGELCHRYKCLLLVDSVASLGGTPLYM
^DRQGIDILYSGSQKALNAPPGTSLISFSDKAKKKMYSRKTKPFSFYLDIKWLANFWGCDDQPRMY
^HHTIPVISLYSLRESLALIAEQGLENSWRQHREAAAYLHGRLQALGLQLFVKDPALRLPTVTTVA
^VPAGYDWRDIVSYVIDHFDIEIMGGLGPSTGKVLRIGLLGCNATRENVDRVTEALRAALQHCPKK
^KL*

N;Alternate names: serine--pyruvate aminotransferase, peroxisomal
N;Contains: alanine--glyoxylate transaminase (EC 2.6.1.44)
C;Species: Homo sapiens (man)
C;Date: 30-Sep-1991 #sequence_revision 30-Sep-1991 #text_change 31-Dec-1993
C;Accession: S10557; A38764; S14002

R;Takada, Y.; Kaneko, N.; Esumi, H.; Purdue, P.E.; Danpure, C.J.
Biochem. J. 268, 517-520, 1990
A;Title: Human peroxisomal L-alanine: glyoxylate aminotransferase. Evolutionary loss
        ^of a mitochondrial targeting signal by point mutation of the initiation
        ^codon.
A;Reference number: S10557; MUID:90303236
A;Accession: S10557
A;Molecule type: mRNA
A;Residues: 1-392 <TAK1>
A;Cross-references: EMBL:X53414
A;Accession: A38764
A;Molecule type: protein
A;Residues: 52-61;318-330 <TAK2>
...
C;Genetics:
A;Gene: GDB:SPAT

C;Complex: homodimer

C;Function:
A;Description: aminotransferase
A;Pathway: glycine biosynthesis

C;Superfamily: serine--pyruvate aminotransferase

F;209/Binding site: #residues Lys #bond_class covalent #ligand pyridoxal phosphate
        ^#status predicted #label BS1
```

Figure 10-1 Sample entry in NBRF format version 6.0

This entry was adapted from PIR-International Protein Sequence Database, Release 41.00, June 1994. Ellipses (...) indicate information omitted for display purposes. The character ^ (which does not occur in the format) is used in this figure to indicate lines that have been wrapped for display on the printed page.

The Header Line corresponds with the ENTRY Record defined for the SDDL representation; it is described in Table 10-1. The *Type* symbols, P1 and F1, of the NBRF format correspond to the *complete* and *fragment* data values of the ENTRY#type field of the SDDL representation. The *CODE* is identical with the ENTRY field value.

Table 10-1 NBRF header line format

Field	Length	Contents of Field
>	1	marks line as an entry header
Type	2	type of sequence in entry **P1** — complete **F1** — fragment
;	1	field separator
CODE	<11	entry identification code

The Title Line consists of free text; it is identical with the data value of the TITLE record of the SDDL representation and is subject to the same stylistic conventions.

The sequence is represented on one or more lines. As in the SDDL representation, it is represented using the one-letter amino acid abbreviations recommended by the IUPAC-IUB Commission on Biochemical Nomenclature [15]. The sequence may contain punctuation characters [2-3,20] and spaces (but no numbering). Generally, the spaces are omitted and the sequence is packed to efficiently fill 500-byte records. The sequence is terminated by an asterisk, *****, that designates the end of the sequence and the Sequence Section.

Records in the Text Section of an entry have the general structure

X;*Identifier***:** data

where: **X** may be **N**, **C**, **R**, **A**, or **F**

N- and C-lines were originally intended to indicate that the data should be formatted as the beginning of a new paragraph. N-lines differ from C-lines only in that they contain information pertaining to descriptions of the protein (alternate names, etc.). R-lines introduce citations to the literature. These lines contain a list of author names and are followed immediately by a line containing the citation. The citation line is the only Text Section line type not introduced by **X;**. A-lines provide information associated with the most closely preceding N-, C-, or R-line (following the citation line). F-lines contain *features* information; they are discussed in Section 13 (and not treated any further in this section). The F-lines occur at the end of the Text Section.

Version 6.0 of the NBRF format introduces extensions to facilitate the interconversion of data between the NBRF format and the SDDL representation. These extensions allow for subidentifiers (and subfields) to be included on N-, C-, or A-lines by preceding them with **#** characters as in the CODATA format [2-3] and in the SDDL representation. N- and C-lines are now interpreted to begin new SDDL records. This provides two mechanisms for expressing the hierarchical grouping of fields defined within SDDL in NBRF format.

NBRF identifiers on A-lines are interpreted as subidentifiers that continue the previous SDDL record. For example,

C;Genetics: *x*
A;Gene: *x*
A;Map_position: *x*

is equivalent to

{ GENETICS *x* #gene *x* #map_position *x* }

The small *x* is used to indicate that the corresponding values are identical.

Alternatively, the subidentifiers may be given explicitly in NBRF format. For example,

C;Date: *x* #sequence_revision *x* #text_change *x*

is equivalent to

{ DATE *x* #sequence_revision *x* #text_change *x* }

The choice of the form used was dictated by stylistic and readability concerns and the desire to mimimize the required changes to previous NBRF format usage. Note that with few exceptions (detailed below) the syntactic formulation of the data within fields and subfields is identical in NBRF version 6.0 and version CO2_6.3 of the SDDL representation. The differences are primarily in the structural aspects of the representation of fields, subfields, and their associated identifiers and subidentifiers.

Identifiers in the SDDL representation are given in all upper-case letters; In NBRF format only the first letter is capitalized. SDDL identifiers and subidentifiers cannot contain space characters; by convention, underscore characters are used to separate words within multiword identfiers. In NBRF format these underscores are replaced by spaces for all record identifiers and for subidentifiers when they introduce an A-line. In SDDL representation, subidentifiers are given in all lower-case letters; in NBRF format, the first character of the subidentifier is capitialized when it occurs as the first element on an A-line. In NBRF format identifiers introducing A-lines are terminated by a colon.

The following lists the Text Section records and their correspondence with SDDL records in their order of occurrence. The symbol $_{\wedge}$ is used to indicate continuation of the same NBRF line.

N;Alternate names: *x*	{ ALTERNATE_NAMES *x* }
N;Contains: *x*	{ CONTAINS *x* }

C;Species: *species_designation*	{ ORGANISM *taxcode*
A;Taxonomy: *taxcode*	#formal_name ... #common_name ...
A;Variety: *x*	#variety *x*
A;Note: *x*	#note *x* }

The *species_designation* contains both the formal and common names of the SDDL representation. Its syntax is described elsewhere [20].

C;Date: *x*	{ DATE *x*
^#sequence_revision *x*	#sequence_revision *x*
^#text_change *x*	#text_change *x* }

C;Accession: *x*	{ ACCESSIONS *x* }

Note that the singular form of the identifier (Accession) is used in NBRF format whereas the plural form (ACCESSIONS) is used in the SDDL representation.

R*; authors*	{ REFERENCE *refnumb*
	#authors *authors*
citation	[#journal, #book, #submission, or #citation]
A;Authors: *authors(continued)*	
[A;Title: *x* or A;Description: *x*]	[##title *x* or ##description *x*]
A;Reference number: *refnumb*;	
^ *crossref*	#cross-references *crossref*
A;Contents: *x*	#contents *x*
A;Note: *x*	#note *x*
A;Accession: *x*	#accession *x*
A;Status: *x*	##status *x*
A;Molecule type: *x*	##molecule_type *x*
A;Residues: *x*	##residues *x*
^ < *label* >	##label *label*
A;Cross-references: *x*	##cross-references *x*
A;Experimental source: *x*	##experimental_source *x*
A;Genetics: *x*	##genetics *x*
A;Note: *x*	##note *x* }

In NBRF format, the author names occur on the R-line. If the author list exceeds the 500-character limit of the R-line, author names are continued over one or more A;Author lines. The syntax of the authors list is identical with the SDDL form.

The original NBRF syntax for *citations* has been retained (it is described elsewhere [20]). The corresponding information is found in the journal, book, submission, or citation subfield of the SDDL REFERENCE record.

The *refnumb* is listed as the first element of the NBRF A;Reference_number line. When present, reference-specific cross-references (from the REFERENCE**#**cross-reference field) follow on this line separated from the *refnumb* by a semicolon.

The accession *label* (from the REFERENCE#accession##label field) is listed at the end of the NBRF A;Residues line enclosed by **<** and **>** characters. As a general rule, unless explicitly denoted by a **#label** subidentifier, SDDL *labels* (SDDL data type _RecordIDY) are enclosed by these characters when they appear in NBRF format.

Note that NBRF format version 6.0 provides direct provision for only one level of field nesting. This context can be retained only by adhering rigorously to the NBRF record order as specified above. Because the #accession##note field cannot occur without the presence of an intervening #accession field, there is no ambiguity in the context of the two NBRF Reference A;Note lines.

C;Comment: *x* ^#link *link* ^< *label* >	{ COMMENT *x* #label *label* #link *link* }
C;Genetics: < *label* > A;Gene: *x* A;Map position: *x* A;Genome: *x* A;Gene origin: *x* A;Genetic code: *x* A;Start codon: *x* A;Introns: *x* ^#status *x* A;Other products: *x* A;Note: *x*	{ GENETICS *label* #gene *x* #map_position *x* #genome *x* #gene_origin *x* #genetic_code *x* #start_codon *x* #introns *x* ##status *x* #other_products *x* #note *x* }
C;Complex: *x*	{ COMPLEX *x* }
C;Function: < *label* > A;Description: *x* A;Pathway: *x* A;Note: *x*	{ FUNCTION *label* #description *x* #pathway *x* #note *x* }
C;Superfamily: *x* A;Group: *x*	{ CLASSIFICATION #superfamily *x* #group *x* }
C;Keywords: *x*	{ KEYWORDS *x* }

The correspondence between NBRF format and SDDL feature lines is discussed in Section 12 (Appendix E). There are no records in NBRF format equivalent to the SUMMARY Record of the SDDL representation.

11 Appendix D

Compatibility with CODATA format version 3.0

The CODATA format is a sequence data exchange format recommended by the CODATA Task Group on the Coordination of Protein Sequence Data Banks [2]. Version 3.0 of the CODATA format is described in PIR Document CXFSD-0494 [3]. CODATA format is a context-independent, free format wherein data are not restricted to specific columns, fields, or records but are identified by a defined set of field descriptors (identifiers and subidentifiers). For maximum portability, the data are represented in the International ASCII character set restricted to the printable ASCII characters (ASCII characters 32 through 126, decimal representation). The data are represented in upper- and lower-case letters; however, no significance is attached to the case of a letter; upper- and lower-case letters are treated equivalently. Records are fixed at 80 characters per record padded on the right by space characters.

The data are organized into entries that contain all the information associated with a particular sequence, including the title, the biological source, references, associated text, and the sequence itself. The database is contained in a single file; the first several lines of the file contain the database header, which identifies the database. The entries follow sequentially; they are separated from any other text in the file by beginning and ending with a record containing three backslash characters, \, in the first three columns.

Types of data within an entry are distinguished by dividing them into specific data items, e.g., title, reference, feature table, etc. Space characters are used as general separators; a variable number of spaces are used to separate data, which allows the data to be represented in an easily readable, tabular form.

Each data item is labeled with an Identifier. Identifiers are single words or several words connected by hyphens or underscores (they contain no internal space characters). A data item may extend over more than one line. The identifier starts at the first column of the first line of the corresponding data item; continuation lines are distinguished by containing at least three space characters at the beginning of the line.

Data within data items are divided into subitems. Each subitem consists of a subidentifier, which identifies the subitems and separates them, followed by the associated data. Subidentifiers are of the same form as identifiers but are immediately preceded by a number sign, **#**, which designates them as subidentifiers; the number sign appears in the database only to introduce subidentifiers. Fields may be nested further by introducing each subfield with a subidentifier preceded by an additional number sign (one for each level of nesting).

The Protein Sequence Component

The representation employed in SDDL is a superset of the CODATA exchange format with the provision that CODATA *data items* are equivalent to SDDL *records*, where CODATA *identifiers* and *subidentifiers* correspond to *record identifiers* and *field labels* of SDDL, respectively. SDDL employs a different record demarcation strategy. In SDDL records are enclosed by braces, **{** and **}**, and physical record delimiters (line-feeds, carriage-controls, etc.) are treated as equivalent to white space. In CODATA, physical record delimiters are significant; they are used for demarcating logical records (data items).

CODATA format versions correspond to fixed database schemas. CODATA version 3.0 reflects the schema associated with the initial implementation of CO2_6.3 of the protein sequence component. Figure 11-1 shows a sample entry in version 3.0 of the CODATA format. With the following exceptions it is identical with the SDDL version except for the record-delimiting conventions. Other differences are found within the feature records; these are described in Section 12 (Appendix E).

1 The #journal, #book, and #submission fields of the REFERENCE record are not parsed and represented as subfields; rather, they retain the syntactic characteristics as specified in the earlier versions of the CODATA format [2,3].

2 The ##title subfields of #journal, #book, and #citation and the ##description subfield of #submission are retained as fields of REFERENCE (i.e., #title and #description, respectively) as in the earlier versions of the CODATA format [2,3].

3 As specified in the earlier versions of the CODATA format [2,3], there is only one type of FEATURE record. Features are distinguished by listing the feature type as a subidentifier rather than as the record identifier. For example, the REGION record is specified in CODATA as **FEATURE ... #region**. The compatibility of feature representations is discussed in more detail in Section 12.

```
ENTRY                XNHUSP                          #type complete
TITLE                serine--pyruvate transaminase (EC 2.6.1.51), peroxisomal
                        - human
ALTERNATE_NAMES      serine--pyruvate aminotransferase, peroxisomal
CONTAINS             alanine--glyoxylate aminotransferase (EC 2.6.1.44)
ORGANISM             #formal_name Homo sapiens #common_name man
DATE                 30-Sep-1991 #sequence_revision 30-Sep-1991
                        #text_change 31-Dec-1993
ACCESSIONS           S10557; A38764; S14002

REFERENCE            S10557
   #authors          Takada, Y.; Kaneko, N.; Esumi, H.; Purdue, P.E.;
                        Danpure, C.J.
   #journal          Biochem. J. (1990) 268:517-520
   #title            Human peroxisomal L-alanine: glyoxylate
                        aminotransferase. Evolutionary loss of a mitochondrial
                        targeting signal by point mutation of the initiation
                        codon.
   #cross-references MUID:90303236
   #accession        S10557 ##molecule_type mRNA  ##residues 1-392
                        ##label TAK1 ##cross-references EMBL:X53414
   #accession        A38764 ##molecule_type protein ##residues 52-61;318-330
                        ##label TAK2
...
GENETICS             #gene GDB:SPAT
COMPLEX              homodimer
FUNCTION             #description aminotransferase #pathway glycine
                        biosynthesis
CLASSIFICATION       #superfamily serine--pyruvate aminotransferase

FEATURES      209    #binding_site #residues Lys  #bond_class covalent
                        #ligand pyridoxal phosphate  #status predicted
                        #label BS1

SUMMARY           #length 392 #molecular_weight 43010 #checksum 1797
SEQUENCE             5          10         15         20         25
            1 M A S H K L L V T P P K A L L K P L S I P N Q L L
            ...
///
```

Figure 11-1 Sample entry in CODATA format version 3.0
This entry was adapted from PIR-International Protein Sequence Database, Release 41.00, June 1994. Ellipses (...) indicate information omitted for display purposes.

12 Appendix E

Features compatibility

SDDL introduced a refinement in the specification of feature locations. Within the SDDL, locations are considered to be objects whose syntax and properties are defined by their class (data type). The locations associated with different feature types, such as products, domains, bonds, etc., correspond to different object classes. The class definition includes a specification of the methods (procedural functions) that decode and manipulate the location syntax (refer to Reference 1 for a more complete discussion). This understanding, which was implicit in the earlier versions of the NBRF and CODATA formats, has been made explicit with the introduction of the SDDL. Consequently, although an effort has been made to retain much of the original representation strategies employed in the NBRF and CODATA formats, the meaning (semantics) of the features has been changed accordingly.

The syntaxes employed in the object classes corresponding to feature locations is a superset of the feature location syntax originally defined for the NBRF and CODATA formats; the latter two forms are identical. NBRF, version 6.0 and CODATA, version 3.0 introduce the full range of feature location specification syntaxes defined by SDDL. They are feature-type specific as defined by the SDDL representation. Refer to the SDDL descriptions for a complete specification of these syntaxes and their usages. A major change introduced in SDDL that is reflected in the revised NBRF and CODATA formats is that location specifications may be null (not present on the feature records).

In NBRF format, all features are represented on F-lines. The correspondence between NBRF F-lines and SDDL feature records is as follows.

F; *location* / **feature descriptor**	{ **record identifier** *location*
^*description*	#description *description*
^*qualifiers*	*qualifiers*
^< *label* >	#label *label* }

F-lines are introduced by **F;** followed immediately by the feature location. The location is separated from the remander of the record by a slash, **/**. Following the slash separator is the feature descriptor that identifies the feature type; it is equivalent to the SDDL record identifier. A decription of the feature follows; in general, this information is identical with that found in the SDDL **#description** field. The description is followed by all other SDDL subfields valid for the corresponding record type. Only the **#label** field is represented

specially, the *label* occurs last on the NBRF record enclosed in angle brackets. The correspondence between NBRF *feature descriptors* and SDDL *record identifiers* is shown in Table 12-1.

Table 12-1 Correspondences among feature type designators

version CO2_6.3 record identifier/field label		**NBRF, version 6.0** feature descriptor	**CODATA,version 3.0** feature descriptor
		--- sequence elements ---	
PRODUCT		Protein:	#protein
PRODUCT		Peptide:	#peptide
DOMAIN		Domain:	#domain
REGION		Region:	#region
		--- bonds ---	
BOND#bond_type	disulfide	Disulfide bonds:	#disulfide_bonds
BOND#bond_type	*description*	Cross-link: *description*	#cross-link *description*
		--- sites ---	
CLEAVAGE_SITE		Cleavage site	#cleavage_site
SITE#class	inhibitory	Inhibitory site:	#inhibitory_site
SITE#class	modified	Modified site:	#modified_site
BINDING_SITE		Binding site:	#binding_site
ACTIVE_SITE		Active site:	#active_site

CODATA features are represented on FEATURES Records. The correspondence of these records to the SDDL feature records is as follows.

FEATURES *location* #**feature_descriptor** *description* *qualifiers*	{ **record identifier** *location* #description *description* *qualifiers* }

The record identifier, FEATURES, is followed immediately by the location. The feature descriptor is given as the first subidentifier, which is followed by a description. In general the description is identical with that found in the SDDL **#description** field. The description is followed by all other SDDL subfields valid for the corresponding record type. The correspondence between CODATA *feature descriptors* and SDDL *record identifiers* is shown in Table 12-1. Note that, like SDDL, the CODATA format permits record concatenation. When this device is employed the FEATURES record identifier may not appear explicitly.

In version CO2_6.3 of the SDDL representation the protein and peptide features have been combined into a single product feature type. NBRF version 6.0 and CODATA 3.0 retain the original distinction; however, this will be discontinued in future versions.

Disulfide bond features are represented on BOND Records in SDDL with **#bond_type** disulfide. Cross-link features are also represented as BOND Records in SDDL. For these features, the **#bond_type** appears as the first term of the *description* on the NBRF and CODATA feature records; these descriptions may be followed by additional text with no intervening punctuation. Consequently, the **#bond_type** can be parsed from NBRF and CODATA feature records only by exact (left-anchored) matching with valid SDDL bond types (as given in Table 6-2).

Inhibitory and modified sites are represented on SITE Records in SDDL with **#class** inhibitory and modified, respectively.

13 Literature

1 George, D.G., Orcutt, B.C., Mewes, H.-W., and Tsugita, A., An object-oriented sequence database definition language (SDDL), *Protein Seq. Data Anal.* **5**, 357-399, 1994

2 George, D.G., Mewes, H.W., and Kihara, H., A standardized format for sequence data exchange, *Protein Seq. Data Anal.* **1**, 27-39, 1987

3 PIR-International Protein Sequence Database: CODATA Exchange Format Specification, PIR Document CXFSD-0494, version 3.0, April 1, 1994

4 Benson, D.A., Boguski, M., Lipman, D.J., and Ostell, J., GenBank, *Nucl. Acids Res.* **22**, 3441-3444, 1994

5 Benson, D., NCBI sequence database program, In: *CODATA bulletin: information integration for biological macromolecules*, Barker, W.C., ed., pp. 76-79, 1991

6 Emmert, D.B., Stoehr, P.J., Stoesser, G., and Cameron, G.N., The European Bioinformatics Institute (EBI) databases, *Nucl. Acids Res.* **22**, 3445-3449, 1994

7 Tateno, Y., Ugawa, Y., Yamazaki, Y., Hayashida, H., Saitou, N., and Gojobori, T., The DNA data bank of Japan, In: *CODATA bulletin: information integration for biological macromolecules*, Barker, W.C., ed., pp. 74-75, 1991

8 Fasman, K.H., Cuticchia, A.J., and Kingsbury, D.T., The GDB™ Human Genome Data Base anno 1994, *Nucl. Acids Res.* **22**, 3462-3469, 1994

9 The FlyBase Consortium, FlyBase—the *Drosophila* database, *Nucl. Acids Res.* **22**, 3456-3458, 1994

10 Dölz R., Mossé, M.-O., Slominski, P.P., Bairoch, A., and Linder, P., LISTA, LISTA-HOP and LISTA-HON: a comprehensive compilation of protein encoding sequences and its associated homology databases from the yeast *Saccharomyces*, *Nucl. Acids Res.* **22**, 3459-3461, 1994

11 Abola, E.E., Bernstein, F.C., Bryant, S.H., Koetzle, T.F., and Weng, J., Protein data bank, In: *Crystallographic databases - information content, software systems, scientific applications*, Allen, F.H., Bergerhoff, G., and Sievers, R., eds., Data Commission of the International Union of Crystallography, Cambridge, 1987

12 Masys, D.R., New directions in bioinformatics, *J. Res. Natl. Inst. Stand. Tech.* **94**, 69-63, 1989

13 French, J.C., Jones, A.K., and Pfaltz, J.L., *Scientific database management: report of the invitational NSF workshop on scientific database management*, Technical Report 90-21, Department of Computer Science, University of Virginia, Charlottesville, VA, 1990

14 Devereux, J., Haeberli, P., and Smithies, O., A comprehensive set of sequence analysis programs for the VAX, *Nucl. Acids Res.* **12**, 387-395, 1984

15 IUPAC-IUB Commission on Biochemical Nomenclature (Kreil, B., Eck, R.V., Dayhoff, M.O., and Cohn, W.E.), A one-letter designation for amino acid sequences: tentative rules, *J. Biol. Chem.* **243**, 3557-3559, 1969

16 Batini, C., Ceri, S., and Navathe, S.B., *Conceptual database design: an entity-relationship approach*, The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, 1992

17 PIR-International Protein Sequence Database: CODATA Exchange Format Specification, PIR Document CXFSD-0694, version 3.0, June, 1994.

18 Orcutt, B.C., George, D.G., Fredrickson, J.A., and Dayhoff, M.O., Nucleic acid sequence database computer system, *Nucl. Acids Res.***10**, 157-174, 1982.

19 Orcutt, B.C., George, D.G., and Dayhoff, M.O., Protein and nucleic acid sequence database systems, *Annu. Rev. Biophys. Bioeng.* **12**, 419-441, 1983

20 PIR-International Protein Sequence Database: Database File Structure and Format Specification, PIR Document PRDBFS-1292, version 5.2, December 1992.