

BioPAX – Biological Pathways Exchange Language

Level 3, Release Candidate 3 (Version 0.92)

Documentation

BioPAX Recommendation, March 17, 2008

The BioPAX data exchange format is the joint work of the BioPAX workgroup and Level 3 builds on the work of Level 2 and Level 1.

BioPAX Level 3 input from: Mirit Aladjem, Ozgun Babur, Gary D. Bader, Burk Braun, Michelle Carrillo, Michael P. Cary, Kei-Hoi Cheung, Julio Collado-Vides, Dan Corwin, Emek Demir, Peter D'Eustachio, Ken Fukuda, Marc Gillespie, Li Gong, Gopal Gopinathrao, Nan Guo, Peter Hornbeck, Michael Hucka, Olivier Hubaut, Geeta Joshi-Tope, Peter Karp, Shiva Krupa, Christian Lemer, Joanne Luciano, Irma Martinez-Flores, Zheng Li, David Merberg, Huaiyu Mi, Nicolas Le Novere, Elgar Pichler, Suzanne Paley, Monica Penaloza-Spinola, Victoria Petri, Elgar Pichler, Alex Pico, Harsha Rajasimha, Ranjani Ramakrishnan, Dean Ravenscroft, Jonathan Rees, Liya Ren, Alan Ruttenberg, Matthias Samwald, Chris Sander, Frank Schacherer, Carl Schaefer, Nigam Shah, Andrea Splendiani, Paul Thomas, Imre Vastrik, Ryan Whaley, Edgar Wingender, Guanming Wu, Jeremy Zucker

BioPAX Level 2 input from: Mirit Aladjem, Gary D. Bader, Ewan Birney, Michael P. Cary, Dan Corwin, Kam Dahlquist, Emek Demir, Peter D'Eustachio, Ken Fukuda, Frank Gibbons, Marc Gillespie, Michael Hucka, Geeta Joshi-Tope, David Kane, Peter Karp, Christian Lemer, Joanne Luciano, Elgar Pichler, Eric Neumann, Suzanne Paley, Harsha Rajasimha, Jonathan Rees, Alan Ruttenberg, Andrey Rzhetsky, Chris Sander, Frank Schacherer, Andrea Splendiani, Lincoln Stein, Imre Vastrik, Edgar Wingender, Guanming Wu, Jeremy Zucker

BioPAX Level 1 input from: Gary D. Bader, Eric Brauner, Michael P. Cary, Emek Demir, Andrew Finney, Ken Fukuda, Robert Goldberg, Susumu Goto, Chris Hogue, Michael Hucka, Peter Karp, Minoru Kanehisa, Stan Letovksy, Joanne Luciano, Debbie Marks, Natalia Maltsev, Elizabeth Marland, Peter Murray-Rust, Eric Neumann, Suzanne Paley, John Pick, Aviv Regev, Andrey Rzhetsky, Chris Sander, Vincent Schachter, Imran Shah, Mustafa Syed, Jeremy Zucker

Thanks to the many additional people who contributed to discussions on the various BioPAX mailing lists and at BioPAX meetings.

This document edited by Gary D. Bader and Emek Demir. Also edited for Level 1 and Level 2 by Michael P. Cary.

Copyright © 2007 BioPAX Workgroup. Some rights reserved under the Creative Commons License (<http://creativecommons.org/licenses/by/2.0/>)

Abstract

At present, there are almost 250 Internet-accessible databases that store biological pathway data. Biologists often need to use information from many of these to support their research, but since each has its own representation conventions and data access methods, integrating data from multiple databases is very difficult. A widely-adopted biological pathway data exchange format will help.

BioPAX (Biological Pathway Exchange - <http://www.biopax.org>) enables the integration of diverse pathway resources by defining an open file format specification for the exchange of biological pathway data. By utilizing the BioPAX format, the problem of data integration reduces to a semantic mapping between the data models of each resource and the data model defined by BioPAX. Widespread adoption of BioPAX for data exchange will increase access to and uniformity of pathway data from varied sources, thus increasing the efficiency of computational pathway research.

This document describes BioPAX Level 3, which expands the scope of BioPAX to include states of physical entities, generic physical entities, gene regulation and genetic interactions. BioPAX Level 3 allows the representation of the bulk of pathway data in publicly available databases.

Scope of this document

This BioPAX documentation is targeted at computational biologists with an interest in biological pathway data. For an overview of BioPAX, read the introduction (section 1). It is expected that readers are familiar with one or more pathway databases and have a basic understanding of both bioinformatics and molecular and cellular biology. This background information is available in a number of textbooks¹.

This document provides an overview the BioPAX Level 3 ontology. This includes descriptions of the BioPAX ontology classes, sample use cases and best practice recommendations. This document does not provide a full definition of the BioPAX Level 3 ontology, which is given by the BioPAX Level 3 OWL file, located:

<http://www.biopax.org/release/biopax-level3.owl>

New Features in BioPAX Level 3

The major change in BioPAX Level 3 is that the representation of physical entities (e.g. proteins) has been redesigned to support physical entities in diverse states and generic physical entities. This has required the removal of some utility classes and the addition of some new ones. Support for new features required backwards incompatible changes compared to the BioPAX Level 1 and 2 formats, however, the majority of the classes and properties are unchanged.

Better support for physical entities in diverse states

A protein, as recorded in a sequence database like UniProt, is now represented as a ReferenceProtein, which stores the protein sequence, name, external references, and potential sequence features (this is similar in meaning to the class 'protein' in BioPAX level 1 and 2). The actual protein chemical species post-translationally modified, bound in a complex or present in a

specific cellular compartment, that participates in an interaction is now represented as the class 'protein' (this is similar in meaning to the class `physicalEntityParticipant` in BioPAX level 1 and 2, except in Level 3 `physicalEntity`'s, stoichiometry is removed and they are no longer duplicated). This new design makes it easier to create different forms of a protein while not duplicating information common to all forms (e.g. protein sequence) and explicitly linking all forms of a protein together (they share the same `EntityReference`). Representation of sequence features and stoichiometry were significantly changed. Most other physical entities, DNA, RNA and small molecule, have similarly been changed. Only complex has not been changed, since it is composed of other physical entities that have been changed.

The `physicalEntityParticipant` class has been removed, as it is no longer needed with the new design. This makes BioPAX easier to use, since now interactions reference their participants directly, not through an intermediate `physicalEntityParticipant` class.

Support for generic physical entities

Generic physical entities are often used in pathway databases e.g. alcohols, nucleotides (dNTPs), and the Wnt protein family (there are many different Wnt genes and proteins in some genomes). Different types of these physical entity groupings are used, such as homolog groups or groups of `SmallMolecules` that share the same chemical functional group. These can now be represented using the `EntityReferenceGroup` class, instances of which can contain multiple `ReferenceEntities` of the same type. Generic features, such as binding sites or post-translational modifications, are also supported.

Support for gene regulation networks

Gene regulation networks, involving regulators of gene expression (e.g. transcription factors, microRNAs) and their targets can now be represented. The new `TemplateReaction` class captures polymerization of macromolecule polymers from a DNA or RNA template. It stores the template, product and regulatory elements. `TemplateReactionRegulation`, involving an expression regulator physical entity (e.g. transcription factor), controls `TemplateReaction`.

Support for genetic interactions

Genetic interactions, such as epistasis or synthetic lethality, are important for mapping pathways, especially in model organisms like yeast, worm, fly and mouse. This information is increasingly available in pathway and interaction databases. To capture these interactions, there is now a `GeneticInteraction` class, which contains a set of genes and a phenotype (expressed using PATO PhenoXML or other phenotype controlled vocabulary). Controlled vocabulary terms to support genetic interactions have also been added to the PSI-MI controlled vocabulary. The gene class has also been added to support genetic interactions.

Support for degradation

Degradation of physical entities, such as proteins, is important in many regulatory pathways. A new `Degradation` conversion class has been added to capture this event. The left side of the interaction contains the degradation substrate and the right side is empty, signifying that the degradation products are not tracked within BioPAX and return to an unspecified molecule pool in the cell.

Major changes from BioPAX Level 2

Warning! The semantics of the physicalEntity classes have changed, but their names have not. E.g. protein now refers to a protein in a state, whereas it used to refer to the base definition of the protein, as would be found in a protein sequence database. This base definition is now a utility class, called EntityReference.

The PathwayStep class has been moved to a new property in pathway to make pathways easier to create (you only need to create pathwayStep instances if you want to order parts of the pathway). Also, there is a new biochemicalPathwayStep class, a subclass of pathwayStep, to make ordering biochemical pathways easier.

physicalInteraction, which stores molecular interactions from e.g. proteomics experiments, has been moved to be a child of the interaction class and named MolecularInteraction. This recognizes that it is a different type of interaction than control and conversion, which were previously children of physicalInteraction.

All controlled vocabulary references now have their own class. E.g. BioSource references TissueVocabulary. This makes use of external controlled vocabularies easier to use. Also, the openControlledVocabulary class has been renamed ControlledVocabulary.

The confidence class has been renamed to score to make it more general, for use in genetic interaction.

Cardinality restrictions that documented required properties and optional properties have been made functional, where possible. More documentation is added to state which functional properties are required vs. optional.

By popular demand, all class names have been changed to the standard CamelCase and all property names to mixedCase.

Other notes

Programmatic access to BioPAX Level 3 is supported by PaxTools, a Java library for reading, writing and validating BioPAX files. PaxTools is available at <http://biopax.org/paxtools>

This document is still a draft. The final version will include graphical representations of the various classes and some examples. Also, this document contains the latest version of the documentation and the BioPAX OWL file does not. When final, the BioPAX OWL file will contain the same documentation as in this file.

Representation Styles Supported in BioPAX Level 3

Different pathway representation styles are in common use for different types of pathway information. Each style is tailored to make representation of the specific type of pathway data easier. Multiple representation styles are supported in BioPAX Level 3.

Metabolic pathways

Metabolic pathways mostly involve biochemical reactions where protein enzymes convert small molecule reactants to small molecule products. While there are many exceptions to this general statement, the majority of metabolic pathway data in databases is covered. BioPAX Level 1 introduced support for this pathway data type.

Molecular interactions

Molecular interactions typically present in proteomics and functional genomics databases involve mainly binary and set interactions between proteins (protein-protein interactions), DNA (protein-DNA interactions) and sometimes other molecules. Experimental description is important for this pathway data type, but otherwise the molecular interactions are known at a low level of detail. BioPAX Level 2 introduced support for this pathway data type.

Signaling Pathways

Signaling pathways mostly involve cascades of protein and other molecule chemical modifications to implement information transfer across the cell. An important difference between these pathways and metabolic or proteomics data is the central role of molecular states, such as protein post-translational modifications, and generic entities, such as the class of Wnt genes. Extensive support of this pathway data type is introduced in BioPAX Level 3.

Gene Regulatory Networks

Gene regulatory networks are composed of regulator-target relationships involved in regulation of gene expression, such as relationships between transcription factors and the genes they regulate. Support for this pathway data type is introduced in BioPAX Level 3.

Genetic Interactions

A genetic interaction takes place when the action of one gene is modified by one or more genes that assort independently. Genetic interactions are used extensively to map pathways in biology. Support for this pathway data type is introduced in BioPAX Level 3.

Key definitions

BioPAX workgroup: Community group designing the BioPAX ontology and format.

BioPAX ontology: The abstract representation of biological pathway concepts and their relationships developed by the BioPAX workgroup. This is also called the object model.

BioPAX format: The file format implementation of the BioPAX ontology that defines the syntax of representation for data. The BioPAX format is currently implemented only in OWL, but other implementations, such as XML Schema may be developed in the future.

OWL: Web Ontology Language. OWL is an XML-based language defined by the World Wide Web Consortium (see <http://www.w3.org/TR/owl-guide/>). OWL can be used to both define an ontology and to store instance data that adheres to that ontology. It is intended that the BioPAX ontology is used to validate that a set of instances follows all BioPAX defined syntax rules. It is

recommended that the BioPAX ontology be imported from its location on the biopax.org website, although it may also be defined directly within an instance data document.

Status of this document

This document is a draft of the final BioPAX Level 3 documentation and is open for comment. Comments on this specification may be sent to biopax-discuss@biopax.org; archives of the comments are available by subscribing to our mailing list here: <http://www.biopax.org/mailman/private/biopax-discuss/>.

Discussion of certain topics is also on the BioPAX wiki at <http://biopaxwiki.org>

This document and the BioPAX Level 3 OWL file will be updated over time, based on community input. The documentation for the latest version of BioPAX Level 3 can always be found here:

<http://www.biopax.org/release/biopax-level3-documentation.pdf>

BioPAX Namespace

The following URI is defined to be the BioPAX Level 3 namespace:

<http://www.biopax.org/release/biopax-level3.owl#>

This namespace name (URI) will always be used to refer to the most recently released version of BioPAX; different URIs will be used for major versions of BioPAX Levels.

Table of contents

BioPAX – Biological Pathways Exchange Language	1
Level 3, Release Candidate 3 (Version 0.92) Documentation	1
Abstract	2
Scope of this document	2
New Features in BioPAX Level 3	2
Better support for physical entities in diverse states	2
Support for generic physical entities	3
Support for gene regulation networks	3
Support for genetic interactions	3
Support for degradation	3
Major changes from BioPAX Level 2	4
Other notes	4
Representation Styles Supported in BioPAX Level 3	4
Metabolic pathways	5
Molecular interactions	5
Signaling Pathways	5
Gene Regulatory Networks	5
Genetic Interactions	5
Key definitions	5
Status of this document	6
BioPAX Namespace	6
Table of contents	7
1 Introduction	11
How to Participate	11
2 BioPAX Ontology Class Structure	13
Top level entity classes	13
Entity (Root class of ontology)	14
Second level classes	15
Pathway	15
Interaction	15
PhysicalEntity	16
Gene	16
Interaction subclasses	17
Summary of Interaction Class Structure	17
Control	17
Conversion	19
GeneticInteraction	20
MolecularInteraction	20
TemplateReaction	20
Control subclasses	21
Catalysis	21
Modulation	22

TemplateReactionRegulation	22
Conversion subclasses	22
BiochemicalReaction	22
ComplexAssembly	24
Degradation	24
Transport	24
TransportWithBiochemicalReaction	24
PhysicalEntity subclasses	25
Complex	25
Dna	26
Protein	26
Rna	26
SmallMolecule	26
Utility classes	26
Top level utility classes	26
BioSource	27
ChemicalStructure	27
ControlledVocabulary	27
DeltaGprime0	28
EntityFeature	29
EntityReference	29
Evidence	31
ExperimentalForm	31
kPrime	31
PathwayStep	33
Provenance	33
Score	34
SequenceLocation	34
Stoichiometry	34
Xref	34
ControlledVocabulary subclasses	35
CellularLocationVocabulary	35
CellVocabulary	35
EntityReferenceGroupVocabulary	35
EvidenceCodeVocabulary	35
ExperimentalFormVocabulary	35
InteractionVocabulary	36
PhenotypeVocabulary	36
SequenceRegionVocabulary	36
SequenceModificationVocabulary	36
TissueVocabulary	36
EntityFeature subclasses	36
BindingFeature	36
ModificationFeature	37
EntityReference subclasses	37
DNAReference	37
ProteinReference	37
RNAReference	38
SmallMoleculeReference	38
PathwayStep subclasses	39
BiochemicalPathwayStep	39

Sequence Location subclasses	39
SequenceInterval	39
SequenceSite	39
Xref subclasses	39
PublicationXref	39
RelationshipXref	40
UnificationXref	40
Summary of BioPAX Class Structure	41
3 Examples	43
4 Best Practices	44
Referencing External Objects	44
Using xrefs	44
Importance of unification xrefs	45
External database identifiers	45
Using external controlled vocabulary terms	46
Cellular location	46
Reusing utility class instances	47
Pathways and networks	47
Black box pathways	47
Pathway ordering	47
Interaction networks	47
Control ‘controller’ and ‘controlled’ property conventions	48
Conversion direction	49
Degradation	49
Conventions for ‘left’ and ‘right’ properties of conversion	49
stepDirection Property of BiochemicalPathwayStep	49
Technical note: OWL and RDF Conventions	50
RDF ID	50
Document namespace	51
5 HOW-TO	53
Creating a knowledge-base using BioPAX and Protégé	53
Viewing Instances Graphically	54
6 Use Case Outlines	55
Data Sharing Between Databases	55
BioPAX as a Knowledge-Base (KB) Model	56
Pathway Data Warehouse	56
Pathway Analysis Software	56
Pathway Analysis Software Example: Molecular profiling analysis	56
Visualizing Pathway Diagrams	57
Pathway Modeling	57
Using BioPAX as metadata for SBML and CellML	57
Pathway analysis using logical inference	57

7 Glossary	59
Appendix A: Design Principles	60
Appendix B: Level and Version Numbers	61
Appendix C: BioPAX Non-Conformance with OWL Semantics	62
Appendix D: Change log	63
References	66

1 Introduction

BioPAX (Biological Pathway Exchange) aims to facilitate the integration and exchange of data maintained in biological pathway databases. Traditionally, integrating data from a number of databases, diverse in form and content, has been a challenge in the field of Bioinformatics². One solution is to define a mutually agreed upon file format as a standard way of representing a given type of data in a community. An example of such a standard is the DDBJ/EMBL/GenBank flat-file format, used to represent nucleic acid sequence data.

Currently, there is no file format standard broadly applicable to biological pathway data, despite the presence of this data in almost 250 different internet accessible databases*. While previous work has been done to standardize specific types of pathway data, such as the successful PSI-MI³ format developed by the protein-protein interaction database community, there is no format capable of representing all of the most frequently used pathway data types. **The goal of the BioPAX project is to provide a data exchange format for pathway data that will represent the key elements of the data models from a wide range of popular pathway databases.** To achieve this goal, the BioPAX ontology was designed to support the data models of a number of existing pathway databases, such as [BioCyc](#)⁴, [BIND](#)⁵, [WIT](#)⁶, [PATIKA](#)⁷, [Reactome](#)⁸, [aMAZE](#)⁹, [KEGG](#)¹⁰, [INOH](#), [NCI/Nature PID](#), [PANTHER Pathways](#)¹¹ and others. When designing the BioPAX ontology, the BioPAX workgroup endeavored to balance the many different representational needs of these and other biological pathway databases.

Because pathway data are complex and can be represented at many levels of detail, the BioPAX group is using a leveled development approach, similar to that of SBML¹². While the overall framework of the BioPAX ontology, i.e. the root class structure, has been designed with the entire pathway data space in mind, representation of specific types of pathway data are the focus of individual levels. **BioPAX Level 1 was designed to represent metabolic pathway data.** Representing other types of pathway data with BioPAX Level 1 is possible but may not be optimal. **BioPAX Level 2 expands the scope of Level 1 to include representation of molecular binding interactions and hierarchical pathways. BioPAX Level 3 adds support for representation of signal transduction pathways, gene regulatory networks and genetic interactions.**

How to Participate

Since a data exchange format is only useful if it is widely adopted, the BioPAX project aims to promote the use of the BioPAX format by as many data sources and consumers as possible. This is achieved partly through community outreach at conferences and workshops, and partly through active participation in the project by data providers and consumers.

We encourage participation in BioPAX! You can help by promoting use of the format, encouraging participation by others, contributing to BioPAX discussions on mailing lists, reviewing BioPAX documents, providing data in the BioPAX format, developing software tools

* <http://www.pathguide.org>

that support the BioPAX format, providing sponsorship for BioPAX activities, participating directly in its design.

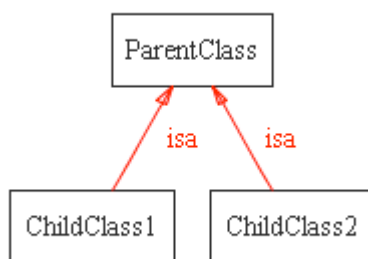
BioPAX participation is currently on a volunteer basis and members have typically paid their own expenses. The US Department of Energy (DOE), Japan's JST and the NIH have provided some funding for holding meetings and will support additional meetings in the future.

More details are available on the www.biopax.org and biopaxwiki.org web sites.

2 BioPAX Ontology Class Structure

This section provides an overview of the BioPAX Level 3 class structure. Full definitions are found in the BioPAX Level 3 OWL document (<http://www.biopax.org/release/biopax-level3.owl>). Text definitions of classes are provided along with synonyms, comments and examples, where possible, to help the reader understand the definition and intended use of each class. If a value is not specified in a property, it is considered unknown. The most specific class available should be used. Effort is made to write clear and concise documentation. However, exact semantics may not always be captured in the class and property documentation. A separate document is being written that will explain class semantics in more detail, where useful, and will be made available with the final documentation.

Interspersed throughout this section are diagrams generated by the Ontoviz Protégé plugin that show the parent/child relationships between selected classes of the BioPAX ontology. In these diagrams, red “isa” arrows connect child classes to their parents.



The sub-property feature of OWL is similar to the subclasses feature. A property of a field may have sub-properties. Any class that contains a property with sub-properties can also contain one of the sub-properties, even if it is not specified. Also, any value of a sub-property will also be a value for the property. For example, the name property has sub-properties for different types of names and the participants property has sub-properties for different participant types, like controller or cofactor.

Top level entity classes

The BioPAX ontology defines 4 basic concepts in the ontology: the root level **Entity** class and four subclasses: **Pathway**, **Interaction**, **PhysicalEntity** and **Gene**.

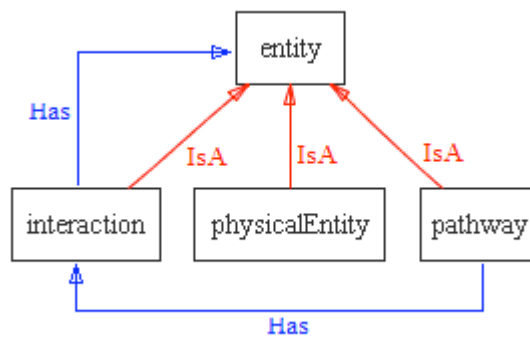


Table 1: Analogies of the root BioPAX ontology structure (first and second level classes) to other conceptual areas.

	Linguistic	Graph representation	Pathway shorthand	Top Level Ontology
PhysicalEntity or Gene	Noun (Subject or Object)	Node	A, B, C	Continuant
Relationship	Verb	Edge	\rightarrow , \Rightarrow	Mediating
Interaction	Phrase/Sentence	Hyperedge, with node labels within context of hyperedge.	$A \rightarrow B$, $B \rightarrow C$	Occurrent
Pathway	Paragraph	Graph	$A \rightarrow B \rightarrow C$	Occurrent

Entity (Root class of ontology)

Definition: A discrete biological unit used when describing pathways.

Comment: This is the root class for all biological concepts in the ontology, which include pathways, interactions and physical entities.

Synonyms: thing, object, bioentity.

Properties:

availability - Describes the availability of this data (e.g. a copyright statement).

comment - Comment on the data in the container class. This property should be used instead of the OWL documentation elements (rdfs:comment) for instances because information in *comment* is data to be exchanged, whereas the rdfs:comment field is used for metadata about the structure of the BioPAX ontology.

dataSource - A description of the source of this data, e.g. a database or person name. This property should be used to describe the source of the data. This should be used by databases that export their data to the BioPAX format or by systems that are integrating data from multiple sources. The granularity of use (specifying the data source in many or few instances) is up to the user. It is intended that this property report the last data source, not all data sources that the data has passed through from creation.

evidence - Scientific evidence supporting the existence of the entity as described.

name - One or more synonyms for the name of this entity. This will automatically include `displayName` and `standardName`, as they are child properties wherever the name property is used.

xref - Values of this property define external cross-references from this entity to entities in external databases.

Second level classes

Pathway

Definition: A set or series of interactions, often forming a network, which biologists have found useful to group together for organizational, historic, biophysical or other reasons.

Comment: It is possible to define a pathway without specifying the interactions within the pathway. In this case, the pathway instance could consist simply of a name and could be treated as a 'black box'.

Synonyms: network

Examples: glycolysis, valine biosynthesis

Properties:

organism - An organism, e.g. 'Homo sapiens'. This is the organism that the pathway is found in. A pathway may not have an organism associated with it, for instance, reference pathways from KEGG.

pathwayComponent - The set of interactions and/or `pathwaySteps` in this pathway/network. Each instance of the `pathwayStep` class defines: 1) a set of interactions that together define a particular step in the pathway, for example a catalysis instance and the conversion that it catalyzes; 2) an order relationship to one or more other pathway steps (via the `NEXT-STEP` property). Note: This ordering is not necessarily temporal - the order described may simply represent connectivity between adjacent steps. Temporal ordering information should only be inferred from the direction of each interaction (see section on biochemical reaction direction in Section 4).

pathwayOrder - The ordering of components (interactions and pathways) in the context of this pathway. This is useful to specific circular or branched pathways or orderings when component biochemical reactions are normally reversible, but are directed in the context of this pathway.

Interaction

Definition: A single biological relationship between two or more entities. An interaction cannot be defined without the entities it relates.

Comment: Currently this class only has subclasses that define physical interactions; later levels of BioPAX may define other types of interactions, such as genetic (e.g. synthetic lethal).

Synonyms: process, synthesis, relationship

Examples: protein-protein interaction, biochemical reaction, enzyme catalysis

Subclasses: Control, Conversion, GeneticInteraction, MolecularInteraction, TemplateReaction

Properties:

interactionType - External controlled vocabulary annotating the interaction type, for example "phosphorylation". This is annotation useful for e.g. display on a web page or database searching, but may not be suitable for other computing tasks, like reasoning.

participant - This property lists the entities that participate in this interaction. For example, in a biochemical reaction, the participants are the union of the reactants and the products of the reaction. This property has a number of sub-properties, such as LEFT and RIGHT used in the BiochemicalInteraction class.

PhysicalEntity

Definition: An entity with a physical structure. A pool of entities, not a specific molecular instance of an entity in a cell.

Comment: This class serves as the super-class for all physical entities, although its current set of subclasses is limited to molecules. This list may be expanded to include photon, environment, cell and cellular component in later levels of BioPAX, depending on community need.

Synonyms: part, interactor, object

Naming rationale: It's difficult to find a name that encompasses all of the subclasses of this class without being too general. E.g. PSI-MI uses 'interactor', BIND uses 'object', BioCyc uses 'chemicals'. physicalEntity seems to be a good name for this specialization of entity.

Examples: protein, small molecule, RNA

Subclasses: Complex, DNA, Protein, RNA, SmallMolecule

Properties:

bindsTo - The physical entities that are non-covalently bound to the referencing physical entity instance.

cellularLocation - A cellular location, e.g. 'cytoplasm'. This should reference a term in the Gene Ontology Cellular Component ontology.

feature - Features of the owner physical entity.

notFeature - Features this owner physical entity is known to be lacking. If not specified, other potential features are not known.

referenceEntity - Reference entity for this physical entity.

Gene

A continuant that encodes information that can be inherited through replication. This is a generalization of the prokaryotic and eukaryotic notion of a gene. This is used only for genetic interactions. Gene expression regulation makes use of DNA and RNA physical entities.

Note: A gene is not a physical entity, but both genes and physical entities are continuants, as defined by most top level ontologies. Gene and PhysicalEntity classes are conceptually similar, though there is no continuant class in BioPAX to group them.

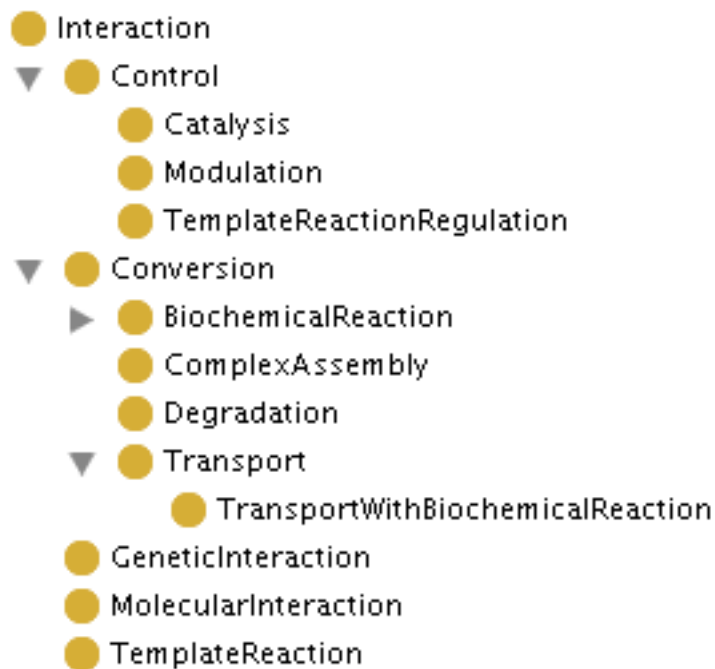
Properties:

organism - An organism, e.g. 'Homo sapiens'. This is the organism that the pathway is found in. A pathway may not have an organism associated with it, for instance, reference pathways from KEGG.

Interaction subclasses

The interaction class has five subclasses: Control, Conversion, GeneticInteraction, MolecularInteraction and TemplateReaction.

Summary of Interaction Class Structure



Control

Definition: An interaction in which one entity regulates, modifies, or otherwise influences another. Two types of control interactions are defined: activation and inhibition.

Comment: In general, the targets of control processes (i.e. occupants of the CONTROLLED property) should be interactions. Conceptually, physical entities are involved in interactions (or events) and the events should be controlled or modified, not the physical entities themselves. For example, a kinase activating a protein is a frequent event in signaling pathways and is usually represented as an 'activation' arrow from the kinase to the substrate in signaling diagrams. This is an abstraction that can be ambiguous out of context. In BioPAX, this information should be captured as the kinase catalyzing (via an instance of the catalysis class) a reaction in which the substrate is phosphorylated, instead of as a control interaction in which the kinase activates the substrate.

Synonyms: regulation, mediation

Examples: A small molecule that inhibits a pathway by an unknown mechanism controls the pathway.

Subclasses: Catalysis, Modulation, TemplateReactionRegulation

Properties:

controlType - Defines the nature of the control relationship between the CONTROLLER and the CONTROLLED entities.

The following terms are possible values:

ACTIVATION: General activation. Compounds that activate the specified enzyme activity by an unknown mechanism. The mechanism is defined as unknown, because either the mechanism has yet to be elucidated in the experimental literature, or the paper(s) curated thus far do not define the mechanism, and a full literature search has yet to be performed.

The following term can not be used in the catalysis class:

INHIBITION: General inhibition. Compounds that inhibit the specified enzyme activity by an unknown mechanism. The mechanism is defined as unknown, because either the mechanism has yet to be elucidated in the experimental literature, or the paper(s) curated thus far do not define the mechanism, and a full literature search has yet to be performed.

The following terms can only be used in the modulation class (these definitions from EcoCyc):

INHIBITION-ALLOSTERIC

Allosteric inhibitors decrease the specified enzyme activity by binding reversibly to the enzyme and inducing a conformational change that decreases the affinity of the enzyme to its substrates without affecting its VMAX. Allosteric inhibitors can be competitive or noncompetitive inhibitors, therefore, those inhibition categories can be used in conjunction with this category.

INHIBITION-COMPETITIVE

Competitive inhibitors are compounds that competitively inhibit the specified enzyme activity by binding reversibly to the enzyme and preventing the substrate from binding. Binding of the inhibitor and substrate are mutually exclusive because it is assumed that the inhibitor and substrate can both bind only to the free enzyme. A competitive inhibitor can either bind to the active site of the enzyme, directly excluding the substrate from binding there, or it can bind to another site on the enzyme, altering the conformation of the enzyme such that the substrate can not bind to the active site.

INHIBITION-IRREVERSIBLE

Irreversible inhibitors are compounds that irreversibly inhibit the specified enzyme activity by binding to the enzyme and dissociating so slowly that it is considered irreversible. For example, alkylating agents, such as iodoacetamide, irreversibly inhibit the catalytic activity of some enzymes by modifying cysteine side chains.

INHIBITION-NONCOMPETITIVE

Noncompetitive inhibitors are compounds that noncompetitively inhibit the specified enzyme by binding reversibly to both the free enzyme and to the enzyme-substrate complex. The inhibitor and substrate may be bound to the enzyme simultaneously and do not exclude each other. However, only the enzyme-substrate complex (not the enzyme-substrate-inhibitor complex) is catalytically active.

INHIBITION-OTHER

Compounds that inhibit the specified enzyme activity by a mechanism that has been characterized, but that cannot be clearly classified as irreversible, competitive, noncompetitive, uncompetitive, or allosteric.

INHIBITION-UNCOMPETITIVE

Uncompetitive inhibitors are compounds that uncompetitively inhibit the specified enzyme activity by binding reversibly to the enzyme-substrate complex but not to the enzyme alone.

ACTIVATION-NONALLOSTERIC

Nonallosteric activators increase the specified enzyme activity by means other than allosteric.

ACTIVATION-ALLOSTERIC

Allosteric activators increase the specified enzyme activity by binding reversibly to the enzyme and inducing a conformational change that increases the affinity of the enzyme to its substrates without affecting its V_{MAX} .

controlled - The entity that is controlled, e.g., in a biochemical reaction, the reaction is controlled by an enzyme. *controlled* is a sub-property of *participant*.

controller - The controlling entity, e.g., in a biochemical reaction, an enzyme is the controlling entity of the reaction. *controller* is a sub-property of *participant*.

Conversion

Definition: An interaction in which one or more entities is physically transformed into one or more other entities.

Comment: This class is designed to represent a simple, single-step transformation. Multi-step transformations, such as the conversion of glucose to pyruvate in the glycolysis pathway, should be represented as pathways, if known.

Examples: A biochemical reaction converts substrates to products, the process of complex assembly converts single molecules to a complex, transport converts entities in one compartment to the same entities in another compartment.

Subclasses: BiochemicalReaction, ComplexAssembly, Degradation, Transport

Properties:

left - The participants on the left side of the conversion interaction. Since conversion interactions may proceed in either the left-to-right or right-to-left direction, occupants of the *left* property may be either reactants or products. *left* is a sub-property of *participant*.

participantStoichiometry - Stoichiometry of the left and right participants.

right - The participants on the right side of the conversion interaction. Since conversion interactions may proceed in either the left-to-right or right-to-left direction, occupants of the *right* property may be either reactants or products. *right* is a sub-property of *participant*.

spontaneous - Specifies whether a conversion occurs spontaneously (i.e. uncatalyzed, under biological conditions) left-to-right, right-to-left, or not at all. If the spontaneity is not known, the *spontaneous* property should be left empty. See the section on reaction direction in Section 4 for how this property can be used to infer direction.

GeneticInteraction

Genetic interactions between genes occur when two genetic perturbations (e.g. mutations) have a combined phenotypic effect not caused by either perturbation alone. This is not a physical interaction, but rather logical. For example, a synthetic lethal interaction occurs when cell growth is possible without either gene A OR B, but not without both gene A AND B. If you knock out A and B together, the cell will die. A gene participant in a genetic interaction represents the gene that is perturbed.

interactionScore - The score of an interaction e.g. a genetic interaction score.

phenotype - The phenotype quality used to define this genetic interaction e.g. viability.

MolecularInteraction

Definition: An interaction with direct molecular contact between entities, but the exact mechanism may not be known.

Comment: This class should be used by default for representing molecular interactions, such as those defined by PSI-MI level 2.5. The participants in a molecular interaction should be listed in the *participant* property. Note that this is one of the few cases in which the *participant* property should be directly populated with instances (see comments on the *participant* property in the interaction class description). If sufficient information on the nature of a molecular interaction is available, a more specific BioPAX interaction class should be used.

Example: Two proteins observed to interact in a yeast-two-hybrid experiment where there is not enough experimental evidence to suggest that the proteins are forming a complex by themselves without any indirect involvement of other proteins. This is the case for most large-scale yeast two-hybrid screens.

TemplateReaction

Definition: This class represents a polymerization of a macromolecule from a template. E.g. DNA to RNA is transcription, RNA to protein is translation and DNA to protein is protein expression from DNA. Other examples are possible. To store a promoter region, create a regulatory element and add a promoter feature on it, using the sequence region vocabulary. The template and

product types determine the type of template reaction. For instance, a DNA template and a protein product is protein expression from DNA, while an RNA template and protein product is protein translation.

product - The product of a template reaction, such as DNA, RNA or protein.

regulatoryElement - A list of regulatory elements involved in a referencing templateReaction instance.

template - The template molecule that is used in this template reaction. Either DNA (e.g. normal gene expression) or RNA (e.g. an RNA virus)

Control subclasses

Three types of control processes exist under the control class: Catalysis, Modulation and TemplateReactionRegulation.

Catalysis

Definition: A control interaction in which a physical entity (a catalyst) increases the rate of a conversion interaction by lowering its activation energy. Instances of this class describe a pairing between a catalyzing entity and a catalyzed conversion.

Comment: A separate catalysis instance should be created for each different conversion that a physicalEntity may catalyze and for each different physicalEntity that may catalyze a conversion. For example, a bifunctional enzyme that catalyzes two different biochemical reactions would be linked to each of those biochemical reactions by two separate instances of the catalysis class. Also, catalysis reactions from multiple different organisms could be linked to the same generic biochemical reaction (a biochemical reaction is generic if it only includes small molecules). Generally, the enzyme catalyzing a conversion is known and the use of this class is obvious. In the cases where a catalyzed reaction is known to occur but the enzyme is not known, a catalysis instance should be created without a controller specified (i.e. the *controller* property should remain empty).

Synonyms: facilitation, acceleration.

Examples: The catalysis of a biochemical reaction by an enzyme, the enabling of a transport interaction by a membrane pore complex, and the facilitation of a complex assembly by a scaffold protein. Hexokinase -> (The “Glucose + ATP -> Glucose-6-phosphate +ADP” reaction). A plasma membrane Na⁺/K⁺ ATPase is an active transporter (antiport pump) using the energy of ATP to pump Na⁺ out of the cell and K⁺ in. Na⁺ from cytoplasm to extracellular space would be described in a transport instance. K⁺ from extracellular space to cytoplasm would be described in a transport instance. The ATPase pump would be stored in a catalysis instance controlling each of the above transport instances. A biochemical reaction that does not occur by itself under physiological conditions, but has been observed to occur in the presence of cell extract, likely via one or more unknown enzymes present in the extract, would be stored in the *controlled* property, with the *controller* property empty.

Properties:

cofactor - Any cofactor(s) or coenzyme(s) required for catalysis of the conversion by the enzyme. *cofactor* is a sub-property of *participant*.

direction - Specifies the reaction direction of the interaction catalyzed by this instance of the catalysis class. Possible values of this property are: REVERSIBLE: Interaction occurs in both directions in physiological settings. PHYSIOL-LEFT-TO-RIGHT PHYSIOL-RIGHT-TO-LEFT The interaction occurs in the specified direction in physiological settings, because of several possible factors including the energetics of the reaction, local concentrations of reactants and products, and the regulation of the enzyme or its expression. IRREVERSIBLE-LEFT-TO-RIGHT IRREVERSIBLE-RIGHT-TO-LEFT For all practical purposes, the interactions occurs only in the specified direction in physiological settings, because of chemical properties of the reaction. (This definition from EcoCyc)

Modulation

Definition: A control interaction in which a physical entity modulates a catalysis interaction. Biologically, most modulation interactions describe an interaction in which a small molecule alters the ability of an enzyme to catalyze a specific reaction. Instances of this class describe a pairing between a modulating entity and a catalysis interaction.

Comment: A separate modulation instance should be created for each different catalysis instance that a physical entity may modulate and for each different physical entity that may modulate a catalysis instance. A typical modulation instance has a small molecule as the controller entity and a catalysis instance as the controlled entity.

Examples: Allosteric activation and competitive inhibition of an enzyme's ability to catalyze a specific reaction.

TemplateReactionRegulation

Definition: Regulation of the expression reaction by the controlling element such as a transcription factor or microRNA. E.g. To represent the binding of the transcription factor to a regulatory element in the TemplateReaction, create a complex of the transcription factor and the regulatory element and set that as the controller.

Conversion subclasses

Five types of conversion processes exist under the conversion class: BiochemicalReaction, ComplexAssembly, Degradation, Transport and TransportWithBiochemicalReaction.

BiochemicalReaction

Definition: A conversion interaction in which one or more entities (substrates) undergo covalent changes to become one or more other entities (products). The substrates of biochemical reactions are defined in terms of sums of species. This is convention in biochemistry, and, in principle, all of the EC reactions should be biochemical reactions.

Examples: $\text{ATP} + \text{H}_2\text{O} = \text{ADP} + \text{P}_i$

Comment: In the example reaction above, ATP is considered to be an equilibrium mixture of several species, namely ATP^{4-} , HATP^{3-} , $\text{H}_2\text{ATP}^{2-}$, MgATP^{2-} , MgHATP^- , and Mg_2ATP . Additional species may also need to be considered if other ions (e.g. Ca^{2+}) that bind ATP are present. Similar considerations apply to ADP and to inorganic phosphate (P_i). When writing

biochemical reactions, it is important not to attach charges to the biochemical reactants and not to include ions such as H⁺ and Mg²⁺ in the equation. The reaction is written in the direction specified by the EC nomenclature system, if applicable, regardless of the physiological direction(s) in which the reaction proceeds. Polymerization reactions involving large polymers whose structure is not explicitly captured should generally be represented as unbalanced reactions in which the monomer is consumed but the polymer remains unchanged, e.g. glucose + glucose = glycogen.

Subclasses: TransportWithBiochemicalReaction

Properties:

deltaG - For biochemical reactions, this property refers to the standard transformed Gibbs energy change for a reaction written in terms of biochemical reactants (sums of species), ΔG° .

$$\Delta G^{\circ} = -RT \ln K' \text{ and } \Delta G^{\circ} = \Delta H^{\circ} - T \Delta S^{\circ}$$

ΔG° has units of kJ/mol. Like K' , it is a function of temperature (T), ionic strength (I), pH, and pMg ($\text{pMg} = -\log_{10}[\text{Mg}^{2+}]$). Therefore, these quantities must be specified, and values for ΔG° for biochemical reactions are represented as 5-tuples of the form (ΔG° T I pH pMg). This property may have multiple values, representing different measurements for ΔG° obtained under the different experimental conditions listed in the 5-tuple. (This definition from EcoCyc)

deltaH - For biochemical reactions, this property refers to the standard transformed enthalpy change for a reaction written in terms of biochemical reactants (sums of species), ΔH° . $\Delta G^{\circ} = \Delta H^{\circ} - T \Delta S^{\circ}$ (This definition from EcoCyc)

deltaS - For biochemical reactions, this property refers to the standard transformed entropy change for a reaction written in terms of biochemical reactants (sums of species), ΔS° . $\Delta G^{\circ} = \Delta H^{\circ} - T \Delta S^{\circ}$ (This definition from EcoCyc)

ecNumber - The unique number assigned to a reaction by the Enzyme Commission of the International Union of Biochemistry and Molecular Biology. Note that not all biochemical reactions currently have EC numbers assigned to them.

keQ - This quantity is dimensionless and is usually a single number. The measured equilibrium constant for a biochemical reaction, encoded by the property KEQ, is actually the apparent equilibrium constant, K' . Concentrations in the equilibrium constant equation refer to the total concentrations of all forms of particular biochemical reactants. For example, in the equilibrium constant equation for the biochemical reaction in which ATP is hydrolyzed to ADP and inorganic phosphate: $K' = [\text{ADP}][\text{P}_i]/[\text{ATP}]$, The concentration of ATP refers to the total concentration of all of the following species: $[\text{ATP}] = [\text{ATP}^{4-}] + [\text{HATP}^{3-}] + [\text{H}_2\text{ATP}^{2-}] + [\text{MgATP}^{2-}] + [\text{MgHATP}^-] + [\text{Mg}_2\text{ATP}]$. The apparent equilibrium constant is formally dimensionless, and can be kept so by inclusion of as many of the terms (1 mol/dm³) in the numerator or denominator as necessary. It is a function of temperature (T), ionic strength (I), pH, and pMg ($\text{pMg} = -\log_{10}[\text{Mg}^{2+}]$). Therefore, these quantities must be specified to be precise, and values for KEQ for biochemical reactions may be represented as 5-tuples of the form (K' T I pH

pMg). This property may have multiple values, representing different measurements for K' obtained under the different experimental conditions listed in the 5-tuple. (This definition adapted from EcoCyc)

ComplexAssembly

Definition: A conversion interaction in which a set of physical entities, at least one being a macromolecule (e.g. protein, RNA, or DNA), aggregate via non-covalent interactions. One of the participants of a complexAssembly must be an instance of the class complex.

Comment: This class is also used to represent complex disassembly. The assembly or disassembly of a complex is often a spontaneous process, in which case the direction of the ComplexAssembly (toward either assembly or disassembly) should be specified via the *spontaneous* property.

Synonyms: aggregation, complex formation

Examples: Assembly of the TFB2 and TFB3 proteins into the TFIID complex, and assembly of the ribosome through aggregation of its subunits.

Note: The following are not examples of complex assembly: Covalent phosphorylation of a protein (this is a BiochemicalReaction); the TFIID complex itself (this is an instance of the complex class, not the ComplexAssembly class).

Degradation

The process of degrading a physical entity. The right side of the conversion is generally not specified, indicating degraded components (see best practices for more details). The conversion does not occur in a right to left direction.

Transport

Definition: A conversion interaction in which an entity (or set of entities) changes location within or with respect to the cell. A transport interaction does not include the transporter entity, even if one is required in order for the transport to occur. Instead, transporters are linked to transport interactions via the catalysis class.

Comment: Transport interactions do not involve chemical changes of the participant(s). These cases are handled by the TransportWithBiochemicalReaction class.

Synonyms: translocation.

Examples: The movement of Na^+ into the cell through an open voltage-gated channel.

Subclasses: TransportWithBiochemicalReaction

TransportWithBiochemicalReaction

Definition: A conversion interaction that is both a BiochemicalReaction and a Transport. In TransportWithBiochemicalReaction interactions, one or more of the substrates change both their location and their physical structure. Active transport reactions that use ATP as an energy source fall under this category, even if the only covalent change is the hydrolysis of ATP to ADP.

Comment: This class was added to support a large number of transport events in pathway databases that have a biochemical reaction during the transport process. It is not expected that other double inheritance subclasses will be added to the ontology at the same level as this class.

Examples: In the PEP-dependent phosphotransferase system, transportation of sugar into an *E. coli* cell is accompanied by the sugar's phosphorylation as it crosses the plasma membrane.

PhysicalEntity subclasses

- PhysicalEntity
 - Complex
 - DNA
 - Protein
 - RNA
 - SmallMolecule

Warning! The semantics of the physicalEntity classes have changed, but their names have not. E.g. protein now refers to a protein in a state, whereas it used to refer to the base definition of the protein, as would be found in a protein sequence database. This base definition is now a utility class, called EntityReference.

Complex

Definition: A physical entity whose structure is comprised of other physical entities bound to each other non-covalently, at least one of which is a macromolecule (e.g. protein, DNA, or RNA). Complexes must be stable enough to function as a biological unit; in general, the temporary association of an enzyme with its substrate(s) should not be considered or represented as a complex. A complex is the physical product of an interaction (complexAssembly) and is not itself considered an interaction.

Comment: In general, complexes should not be defined recursively so that smaller complexes exist within larger complexes, i.e. a complex should not be a COMPONENT of another complex (see comments on the COMPONENT property below). The boundaries on the size of complexes described by this class are not defined here, although elements of the cell as large and dynamic as, e.g., a mitochondrion would typically not be described using this class (later versions of this ontology may include a cellularComponent class to represent these). The strength of binding cannot be described currently, but may be included in future versions of the ontology, depending on community need.

Examples: Ribosome, RNA polymerase II. Other examples of this class include complexes of multiple protein monomers and complexes of proteins and small molecules.

Properties:

component - Defines the PhysicalEntity subunits of this complex. This property should not contain other complexes, i.e. it should always be a flat representation of the complex. For example, if two protein complexes join to form a single larger complex via a complex assembly interaction, the *component* of the new complex should be the individual proteins of the smaller complexes, not the two smaller complexes themselves. Exceptions are black-box complexes (i.e. complexes in which the *component* property is empty), which may be used as *component* of other complexes because their constituent parts are unknown / unspecified. The reason for keeping complexes flat is to signify that there is no information stored in the way complexes are nested, such as assembly order. Otherwise, the complex assembly order may be implicitly encoded and interpreted by some users, while others created hierarchical complexes randomly, which could lead to data loss.

componentStoichiometry - The stoichiometry of components in a complex.

Dna

Definition: A physical entity consisting of a sequence of deoxyribonucleotide monophosphates; a deoxyribonucleic acid.

Comment: This is not a 'gene', since gene is a genetic concept, not a physical entity. The concept of a gene may be added later in BioPAX.

Examples: a chromosome, a plasmid. A specific example is chromosome 7 of Homo sapiens.

Protein

Definition: A physical entity consisting of a sequence of amino acids; a protein monomer; a single polypeptide chain.

Examples: The epidermal growth factor receptor (EGFR) protein.

Rna

Definition: A physical entity consisting of a sequence of ribonucleotide monophosphates; a ribonucleic acid.

Examples: messengerRNA, microRNA, ribosomalRNA. A specific example is the let-7 microRNA.

SmallMolecule

Definition: A small bioactive molecule. Small is not precisely defined, but includes all metabolites and most drugs and does not include large polymers, including complex carbohydrates.

Comment: A number of small molecule databases are available to cross-reference from this class, such as PubChem.

Examples: glucose, penicillin, phosphatidylinositol

Note: Complex carbohydrates are not currently modeled in BioPAX or most pathway databases, due to the lack of a publicly available complex carbohydrate database.

Utility classes

A number of properties in the ontology accept instances of utility classes as values.

Organizational classes in the utility class tree are present to partition the utility class hierarchy into easily navigable subdivisions. Utility classes are created when simple properties are insufficient to describe an aspect of an entity. The UtilityClass class is technically a metaclass and is only present to organize the other helper classes under one class hierarchy.

Top level utility classes

There are 15 direct subclasses of utilityClass: BioSource, **ChemicalStructure**, **ControlledVocabulary**, **DeltaGprime0**, **EntityFeature**, **EntityReference**, **Evidence**, **ExperimentalForm**, **kPrime**, **PathwayStep**, **Provenance**, **Score**, **SequenceLocation**, **Stoichiometry**, and **Xref**.

BioSource

Definition: The biological source of an entity (e.g. protein, RNA or DNA). Some entities are considered source-neutral (e.g. small molecules), and the biological source of others can be deduced from their constituents (e.g. complex, pathway).

Examples: HeLa cells, human, and mouse liver tissue.

Properties:

cellType - A cell type, e.g. 'HeLa'. This should reference a term in a controlled vocabulary of cell types. See the section on controlled vocabularies in Section 4 for more information.

name - One or more synonyms for the name of this entity. This will automatically include *displayName* and *standardName*, as they are child properties wherever the name property is used.

taxonXref - An xref to an organism taxonomy database, preferably NCBI taxon. This should be an instance of *unificationXref*, unless the organism is not in an existing database.

tissue - An external controlled vocabulary of tissue types. See the section on controlled vocabularies in Section 4 for more information.

ChemicalStructure

Definition: Describes a small molecule structure. Structure information is stored in the property *STRUCTURE-DATA*, in one of three formats: the CML format¹³ (see URL www.xml-cml.org), the SMILES format¹⁴ (see URL www.daylight.com/dayhtml/smiles/) or the InChI format (<http://www.iupac.org/inchi/>). The *STRUCTURE-FORMAT* property specifies which format is used.

Comment: By virtue of the expressivity of CML, an instance of this class can also provide additional information about a small molecule, such as its chemical formula, names, and synonyms, if CML is used as the structure format.

Examples: The following SMILES string, which describes the structure of glucose-6-phosphate:

```
'C(OP(=O)(O)O)[CH]1([CH](O)[CH](O)[CH](O)[CH](O)O1)'
```

Properties:

structureData - This property holds a string of data defining chemical structure or other information, in either the CML or SMILES format, as specified in property *Structure-Format*. If, for example, the CML format is used, then the value of this property is a string containing the XML encoding of the CML data.

structureFormat - This property specifies which format is used to define chemical structure data.

ControlledVocabulary

Definition: Used to import terms from external controlled vocabularies (CVs) into the ontology. To support consistency and compatibility, open, freely available CVs should be used whenever possible, such as the Gene Ontology (GO)¹⁵ or other open biological CVs listed on the OBO

website (<http://obo.sourceforge.net/>). See the section on controlled vocabularies in Section 4 for more information.

Comment: The ID property in unification xrefs to GO and other OBO ontologies should include the ontology name in the ID property (e.g. ID="GO:0005634" instead of ID="0005634").

Subclasses: CellularLocationVocabulary, CellVocabulary, EvidenceCodeVocabulary, ExperimentalFormVocabulary, InteractionVocabulary, PhenotypeVocabulary, EntityReferenceGroupVocabulary, SequenceModificationVocabulary, SequenceRegionVocabulary, TissueVocabulary

Properties:

term - The external controlled vocabulary term.

xref - Values of this property define external cross-references from this entity to entities in external databases.

DeltaGprime0

Definition: For biochemical reactions, this property refers to the standard transformed Gibbs energy change for a reaction written in terms of biochemical reactants (sums of species), delta-G'o.

$$\text{delta-G'o} = -RT \ln K'$$

and

$$\text{delta-G'o} = \text{delta-H'o} - T \text{delta-S'o}$$

delta-G'o has units of kJ/mol. Like K', it is a function of temperature (T), ionic strength (I), pH, and pMg (pMg = $-\log_{10}[\text{Mg}^{2+}]$). Therefore, these quantities must be specified, and values for DELTA-G for biochemical reactions are represented as 5-tuples of the form (delta-G'o T I pH pMg). This property may have multiple values, representing different measurements for delta-G'o obtained under the different experimental conditions listed in the 5-tuple.

(This definition from EcoCyc)

deltaGPrimeO - For biochemical reactions, this property refers to the standard transformed Gibbs energy change for a reaction written in terms of biochemical reactants (sums of species), delta-G'o.

$$\text{delta-G'o} = -RT \ln K'$$

and

$$\text{delta-G'o} = \text{delta-H'o} - T \text{delta-S'o}$$

delta-G'o has units of kJ/mol. Like K', it is a function of temperature (T), ionic strength (I), pH, and pMg (pMg = $-\log_{10}[\text{Mg}^{2+}]$). Therefore, these quantities must be specified, and values for DELTA-G for biochemical reactions are represented as 5-tuples of the form (delta-G'o T I pH pMg).

(This definition from EcoCyc)

ionicStrength - The ionic strength is defined as half of the total sum of the concentration (ci) of every ionic species (i) in the solution times the square of its charge (zi). For example, the ionic strength of a 0.1 M solution of CaCl₂ is $0.5 \times (0.1 \times 2^2 + 0.2 \times 1^2) = 0.3 \text{ M}$
(Definition from <http://www.lsbu.ac.uk/biology/enztech/ph.html>)

ph - a measure of acidity and alkalinity of a solution that is a number on a scale on which a value of 7 represents neutrality and lower numbers indicate increasing acidity and higher numbers increasing alkalinity and on which each unit of change represents a tenfold change in acidity or alkalinity and that is the negative logarithm of the effective hydrogen-ion concentration or hydrogen-ion activity in gram equivalents per liter of the solution. (Definition from Merriam-Webster Dictionary)

pMg - A measure of the concentration of magnesium (Mg) in solution. ($pMg = -\log_{10}[Mg^{2+}]$)

temperature - Temperature in Celsius

EntityFeature

A feature or aspect of a physical entity that can be changed while the entity still retains its biological identity.

Subclasses: BindingFeature, ModificationFeature

Properties:

evidence - Scientific evidence supporting the existence of the entity as described.

featureLocation - Location of the feature on the sequence of the interactor. One feature may have more than one location, used e.g. for features which involve sequence positions close in the folded, three-dimensional state of a protein, but non-continuous along the sequence.

EntityReference

Definition: A reference entity is a grouping of several physical entities across different contexts and molecular states that share common physical properties and are often named and treated as a single entity with multiple states by biologists.

Reference entities store the information common to a set of molecules in various states described in the BioPAX document, including database cross-references. For instance, the P53 protein can be phosphorylated in multiple different ways. Each separate P53 protein (pool) in a phosphorylation state would be represented as a different Protein (child of PhysicalEntity) and all things common to all P53 proteins, including all possible phosphorylation sites, the sequence common to all of them and common references to protein databases containing more information about P53 would be stored in a EntityReference. The reference entity is important to link various protein objects representing different states of the same molecule. Also, reference entity allows the creation of unification links to source databases, for instance linking P53 to the UniProt P53 protein record.

Comment: Many protein, small molecule and gene databases share this point of view, and such a grouping is an important prerequisite for interoperability with those databases. Biologists would often group different pools of molecules in different contexts under the same name. For example cytoplasmic and extracellular calcium have different effects on the cell's behavior, but they are still called calcium. This grouping has three semantic implications:

1. Members of different pools share many physical and biochemical properties. This includes their chemical structure, sequence, organism and set of molecules they react with. They will also share a lot of secondary information such as their names, functional groupings, annotation terms and database identifiers.

2. A small number of transitions separates these pools. In other words it is relatively easy and frequent for a molecule to transform from one physical entity to another that belongs to the same reference entity. For example, an extracellular calcium ion can become cytoplasmic, and p53 protein can become phosphorylated. However no calcium ion virtually becomes sodium, or no p53 becomes mdm2. In the former it is the sheer energy barrier of a nuclear reaction, in the latter sheer statistical improbability of synthesizing the same sequence without a template. If one thinks about the biochemical network as molecules transforming into each other, and remove edges that respond to transcription, translation, degradation and covalent modification of small molecules, each remaining component is a reference entity.

3. Some of the pools in the same group can overlap. p53-p@ser15 can overlap with p53-p@thr18. Most of the experiments in molecular biology will only investigate one feature, rarely multiple, and almost never all possible combinations. So almost all statements that refer to the state of the molecule discuss a pool that can overlap with other pools. However no overlap is possible between molecules of different groups.

Subclasses: ReferenceDNA, ReferenceProtein, ReferenceRNA, ReferenceSmallMolecule
Properties:

entityFeature - Variable features that are observed for this entity - such as known PTM or methylation sites and non-covalent bonds.

evidence - Scientific evidence supporting the existence of the entity as described.

groupType – A controlled vocabulary term that is used to describe the type of grouping such as homology or functional group

memberEntity - A reference entity that qualifies for the definition of this group. For example a member of a PFAM protein family.

name - One or more synonyms for the name of this entity. This will automatically include displayName and standardName, as they are child properties wherever the name property is used.

Evidence

Definition: The support for a particular assertion, such as the existence of an interaction or pathway. At least one of CONFIDENCE, EVIDENCE-CODE, or EXPERIMENTAL-FORM must be instantiated when creating an evidence instance. XREF may reference a publication describing the experimental evidence using a publicationXref or may store a description of the experiment in an experimental description database using a unificationXref (if the referenced experiment is the same) or relationshipXref (if it is not identical, but similar in some way e.g. similar in protocol). Evidence is meant to provide more information than just an xref to the source paper.

Examples: A description of a molecular binding assay that was used to detect a protein-protein interaction.

Properties:

confidence - Confidence in the containing instance. Usually a statistical measure.

evidenceCode - A pointer to a term in an external controlled vocabulary, such as the GO or BioCyc evidence codes, that describes the nature of the support. See the section on controlled vocabularies in Section 4 for more information.

experimentalForm - The experimental forms associated with an evidence instance.

xref - Values of this property define external cross-references from this entity to entities in external databases.

ExperimentalForm

Definition: The form of a physical entity in a particular experiment, as it may be modified for purposes of experimental design. This is not the same as biologically relevant modifications captured in physical entity features.

Examples: A His-tagged protein in a binding assay. A protein can be tagged by multiple tags, so can have more than 1 experimental form type terms.

Properties:

experimentalFormDescription - Descriptor of this experimental form from a controlled vocabulary. See the section on controlled vocabularies in Section 4 for more information.

experimentalFormEntity - The Gene or PhysicalEntity that has the experimental form being described.

experimentalFeature - A feature of the experimental form of the participant of the interaction, such as a protein tag. It is not expected to occur in vivo or be necessary for the interaction.

kPrime

Definition: The apparent equilibrium constant, K' , and associated values. Concentrations in the equilibrium constant equation refer to the total concentrations of all forms of particular

biochemical reactants. For example, in the equilibrium constant equation for the biochemical reaction in which ATP is hydrolyzed to ADP and inorganic phosphate:

$$K' = \frac{[\text{ADP}][\text{Pi}]}{[\text{ATP}]},$$

The concentration of ATP refers to the total concentration of all of the following species:

$$[\text{ATP}] = [\text{ATP}^{4-}] + [\text{HATP}^{3-}] + [\text{H}_2\text{ATP}^{2-}] + [\text{MgATP}^{2-}] + [\text{MgHATP}^{-}] + [\text{Mg}_2\text{ATP}].$$

The apparent equilibrium constant is formally dimensionless, and can be kept so by inclusion of as many of the terms (1 mol/dm³) in the numerator or denominator as necessary. It is a function of temperature (T), ionic strength (I), pH, and pMg (pMg = -log₁₀[Mg²⁺]). Therefore, these quantities must be specified to be precise, and values for KEQ for biochemical reactions may be represented as 5-tuples of the form (K' T I pH pMg). This property may have multiple values, representing different measurements for K' obtained under the different experimental conditions listed in the 5-tuple. (This definition adapted from EcoCyc)

See <http://www.chem.qmul.ac.uk/iubmb/thermod/> for a thermodynamics tutorial.

ionicStrength - The ionic strength is defined as half of the total sum of the concentration (c_i) of every ionic species (i) in the solution times the square of its charge (z_i). For example, the ionic strength of a 0.1 M solution of CaCl₂ is 0.5 x (0.1 x 2² + 0.2 x 1²) = 0.3 M (Definition from <http://www.lsbu.ac.uk/biology/enztech/ph.html>)

kPrime - The apparent equilibrium constant K'. Concentrations in the equilibrium constant equation refer to the total concentrations of all forms of particular biochemical reactants. For example, in the equilibrium constant equation for the biochemical reaction in which ATP is hydrolyzed to ADP and inorganic phosphate:

$$K' = \frac{[\text{ADP}][\text{Pi}]}{[\text{ATP}]},$$

The concentration of ATP refers to the total concentration of all of the following species:

$$[\text{ATP}] = [\text{ATP}^{4-}] + [\text{HATP}^{3-}] + [\text{H}_2\text{ATP}^{2-}] + [\text{MgATP}^{2-}] + [\text{MgHATP}^{-}] + [\text{Mg}_2\text{ATP}].$$

The apparent equilibrium constant is formally dimensionless, and can be kept so by inclusion of as many of the terms (1 mol/dm³) in the numerator or denominator as necessary. It is a function of temperature (T), ionic strength (I), pH, and pMg (pMg = -log₁₀[Mg²⁺]). (Definition from EcoCyc)

ph - a measure of acidity and alkalinity of a solution that is a number on a scale on which a value of 7 represents neutrality and lower numbers indicate increasing acidity and higher numbers increasing alkalinity and on which each unit of change represents a tenfold change in acidity or alkalinity and that is the negative logarithm of the effective hydrogen-ion concentration or hydrogen-ion activity in gram equivalents per liter of the solution. (Definition from Merriam-Webster Dictionary)

pMg - A measure of the concentration of magnesium (Mg) in solution. ($pMg = -\log_{10}[Mg^{2+}]$)

temperature - Temperature in Celsius

PathwayStep

Definition: A step in a pathway.

Comment: Multiple interactions may occur in a pathway step, each should be listed in the STEP-INTERACTIONS property. Order relationships between pathway steps may be established with the NEXT-STEP slot. This order may not be temporally meaningful for specific steps, such as for a pathway loop or a reversible reaction, but represents a directed graph of step relationships that can be useful for describing the overall flow of a pathway, as may be useful in a pathway diagram.

Example: A metabolic pathway may contain a pathway step composed of one biochemical reaction (BR1) and one catalysis (CAT1) instance, where CAT1 describes the catalysis of BR1. The M phase of the cell cycle, defined as a pathway, precedes the G1 phase, also defined as a pathway.

Subclasses: BiochemicalPathwayStep

Properties:

evidence - Scientific evidence supporting the existence of the entity as described.

nextStep - The next step(s) of the pathway. Contains zero or more pathwayStep instances. If there is no next step, this property is empty. Scientific evidence supporting the existence of the entity as described.

stepProcess - An interaction or a pathway that are a part of this pathway step.

Provenance

Definition: The direct source of this data. This does not store the trail of sources from the generation of the data to this point, only the last known source, such as a database. The XREF property may contain a publicationXref referencing a publication describing the data source (e.g. a database publication). A unificationXref may be used e.g. when pointing to an entry in a database of databases describing this database.

Examples: A database or person name.

Properties:

name – One or more synonyms for the name of this entity. This will automatically include displayName and standardName, as they are child properties wherever the name property is used.

xref - Values of this property define external cross-references from this entity to entities in external databases.

Score

Definition: A score associated with a publication reference describing how the score was determined, the name of the method and a comment briefly describing the method. The xref must contain at least one publication that describes the method used to determine the score value.

There is currently no standard way of describing values, so any string is valid.

Examples: The statistical significance of a result, e.g. "p<0.05".

Properties:

scoreSource – The name and source of the score, such as a publication describing the scoring system.

value - The value of the score measure.

SequenceLocation

Definition: A location on a nucleotide or amino acid sequence.

Subclasses: SequenceInterval, SequenceLocationGroup, SequenceSite

Properties:

locationType - A controlled vocabulary term describing the type of the sequence location such as C-Terminal or SH2 Domain.

memberLocation - A sequence location that belongs to this homology grouping. Example a particular SH2 domain on a protein.

Stoichiometry

Definition: Stoichiometric coefficient of a physical entity in the context of an interaction or complex. Note that this class is an n-ary specifier for left and right properties.

Properties:

physicalEntity - The physical entity annotated with the stoichiometry attribute from the corresponding Stoichiometry instance.

stoichiometricCoefficient - Each value of this property represents the stoichiometric coefficient for one of the entities in an interaction or complex. For a given interaction, the stoichiometry should always be used where possible instead of representing the number of participants with separate instances of each participant. If there are three ATP molecules, one ATP molecule should be represented as a participant and the stoichiometry should be set to 3.

Xref

Definition: A reference from an instance of a class in this ontology to an object in an external resource.

Subclasses: PublicationXref, RelationshipXref, UnificationXref

Properties:

db - The name of the external database to which this xref refers.

dbVersion - The version of the external database in which this xref was last known to be valid. Resources may have recommendations for referencing dataset versions. For instance, the Gene Ontology recommends listing the date the GO terms were downloaded.

id - The primary identifier in the external database of the object to which this xref refers.

idVersion - The version number of the identifier (ID). E.g. The RefSeq accession number NM_005228.3 should be split into NM_005228 as the ID and 3 as the ID-VERSION.

ControlledVocabulary subclasses

CellularLocationVocabulary

A reference to the Gene Ontology Cellular Component (GO CC) ontology. Homepage at <http://www.geneontology.org>. Browse at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=GO>

CellVocabulary

A reference to the Cell Type Ontology (CL). Homepage at <http://obofoundry.org/cgi-bin/detail.cgi?cell>. Browse at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=CL>

EntityReferenceGroupVocabulary

A reference to a term from a reference entity group ontology.

EvidenceCodeVocabulary

A reference to the PSI Molecular Interaction ontology (MI) experimental method types, including "interaction detection method", "participant identification method", "feature detection method". Homepage at <http://www.psidev.info/>. Browse at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI>

Terms from the Pathway Tools Evidence Ontology may also be used. Homepage <http://brg.ai.sri.com/evidence-ontology/>

ExperimentalFormVocabulary

A reference to the PSI Molecular Interaction ontology (MI) participant identification method (e.g. mass spectrometry), experimental role (e.g. bait, prey), experimental preparation (e.g. expression level) type. Homepage at <http://www.psidev.info/>. Browse

<http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI&termId=MI%3A0002&termName=participant%20identification%20method>

<http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI&termId=MI%3A0495&termName=experimental%20role>

<http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI&termId=MI%3A0346&termName=experimental%20preparation>

InteractionVocabulary

A reference to the PSI Molecular Interaction ontology (MI) interaction type. Homepage at <http://www.psidev.info/>. Browse at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI&termId=MI%3A0190&termName=interaction%20type>

PhenotypeVocabulary

The phenotype measured in the experiment e.g. growth rate or viability of a cell. This is only the type, not the value e.g. for a synthetic lethal interaction, the phenotype is viability, specified by ID: PATO:0000169, "viability", not the value (specified by ID: PATO:0000718, "lethal (sensu genetics)". A single term in a phenotype controlled vocabulary can be referenced using the xref, or the PhenoXML describing the PATO EQ model phenotype description can be stored as a string in *patoData*.

Properties:

patoData - The phenotype data from PATO, formatted as PhenoXML (defined at <http://www.fruitfly.org/~cjm/obd/formats.html>)

SequenceRegionVocabulary

A reference to the Sequence Ontology (SO). Homepage at <http://www.sequenceontology.org/>. Browse at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=SO>

SequenceModificationVocabulary

A reference to the PSI Molecular Interaction ontology (MI) of covalent sequence modifications. Homepage at <http://www.psidev.info/>. Browse at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI&termId=MI%3A0252&termName=biological%20feature>. Only children that are covalent modifications at specific positions can be used.

TissueVocabulary

A reference to the BRENDA (BTO). Homepage at <http://www.brenda-enzymes.info/>. Browse at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=BTO>

EntityFeature subclasses

BindingFeature

Specifies the binding domains of two entities in a complex that are non-covalently bound to each other. Note that this is a n-ary specifier class on the *boundTo* property. The difference between this class and *modificationFeature* is that this is non-covalent and *modificationFeature* is covalent.

Properties:

boundTo – An entity that is non-covalently bound to this entity within the context of the respective protein. All bound physical entity must be in the same complex.

locationType - A controlled vocabulary term describing the type of the sequence location such as C-Terminal or SH2 Domain.

ModificationFeature

Definition: A covalently modified feature on a sequence relevant to an interaction, such as a post-translational modification. The difference between this class and *bindingFeature* is that this is covalent and *bindingFeature* is non-covalent.

Examples: A phosphorylation on a protein.

Properties:

modificationType - Description and classification of the feature.

EntityReference subclasses

DNAReference

A grouping of several DNA entities that are common in sequence and genomic position. Members can differ in cellular location, sequence features, SNPs, mutations and bound partners.

Comments : Note that this is not a reference gene. A gene can possibly span multiple DNA molecules, sometimes even across chromosomes due to regulatory regions. Similarly a gene is not necessarily made up of deoxyribonucleic acid and can be present in multiple copies (which are different DNA molecules).

Properties:

organism - An organism, e.g. 'Homo sapiens'. This is the organism that the entity is found in. Pathways may not have an organism associated with them, for instance, reference pathways from KEGG. Sequence-based entities (DNA, protein, RNA) may contain an xref to a sequence database that contains organism information, in which case the information should be consistent with the value for ORGANISM.

sequence - Polymer sequence in uppercase letters. For DNA, usually A,C,G,T letters representing the nucleosides of adenine, cytosine, guanine and thymine, respectively; for RNA, usually A, C, U, G; for protein, usually the letters corresponding to the 20 letter IUPAC amino acid code.

ProteinReference

A reference protein is a grouping of several protein entities that are encoded by the same gene. Members can differ in cellular location, sequence features and bound partners. Currently conformational states (such as open and closed) are not covered.

Properties:

organism - An organism, e.g. 'Homo sapiens'. This is the organism that the entity is found in. Pathways may not have an organism associated with them, for instance, reference pathways from KEGG. Sequence-based entities (DNA, protein, RNA) may contain an xref to a sequence database that contains organism information, in which case the information should be consistent with the value for ORGANISM.

sequence - Polymer sequence in uppercase letters. For DNA, usually A,C,G,T letters representing the nucleosides of adenine, cytosine, guanine and thymine, respectively; for RNA, usually A, C, U, G; for protein, usually the letters corresponding to the 20 letter IUPAC amino acid code.

RNAReference

A reference RNA is a grouping of several RNA entities that are either encoded by the same gene or replicates of the same genome. Members can differ in cellular location, sequence features and bound partners. Currently conformational states (such as hairpin) are not covered.

Properties:

organism - An organism, e.g. 'Homo sapiens'. This is the organism that the entity is found in. Pathways may not have an organism associated with them, for instance, reference pathways from KEGG. Sequence-based entities (DNA, protein, RNA) may contain an xref to a sequence database that contains organism information, in which case the information should be consistent with the value for ORGANISM.

sequence - Polymer sequence in uppercase letters. For DNA, usually A,C,G,T letters representing the nucleosides of adenine, cytosine, guanine and thymine, respectively; for RNA, usually A, C, U, G; for protein, usually the letters corresponding to the 20 letter IUPAC amino acid code.

SmallMoleculeReference

A reference small molecule is a grouping of several small molecule entities that have the same chemical structure. Members can differ in cellular location and bound partners. Covalent modifications of small molecules are not considered as state changes but treated as different molecules.

Properties:

chemicalFormula – The chemical formula of the small molecule. Note: chemical formula can also be stored in the STRUCTURE property (in CML). In case of disagreement between the value of this property and that in the CML file, the CML value takes precedence.

molecularWeight – Defines the molecular weight of the molecule, in daltons.

structure - Defines the chemical structure and other information about this molecule, using an instance of class chemicalStructure.

PathwayStep subclasses

BiochemicalPathwayStep

Definition: Imposes ordering on a step in a biochemical pathway. A biochemical reaction can be reversible by itself, but can be physiologically directed in the context of a pathway, for instance due to flux of reactants and products. Only one conversion interaction can be ordered at a time, but multiple catalysis or modulation instances can be part of one step.

Properties:

stepConversion - The central process that take place at this step of the biochemical pathway.

stepDirection - Direction of the conversion in this particular pathway context.

stepProcess - An interaction or a pathway that are a part of this pathway step.

Sequence Location subclasses

SequenceInterval

Definition: Describes an interval on a sequence. All of the sequence from the begin site to the end site (inclusive) is described, not any subset.

Properties:

sequenceIntervalBegin - The begin position of a sequence interval.

sequenceIntervalEnd - The end position of a sequence interval.

SequenceSite

Definition: Describes a site on a sequence, i.e. the position of a single nucleotide or amino acid.

Properties:

positionStatus - The confidence status of the sequence position. This could be:

EQUAL: The SEQUENCE-POSITION is known to be at the SEQUENCE-POSITION.

GREATER-THAN: The site is greater than the SEQUENCE-POSITION.

LESS-THAN: The site is less than the SEQUENCE-POSITION.

sequencePosition - The integer listed gives the position. The first base or amino acid is position 1. In combination with the numeric value, the property 'POSITION-STATUS' allows to express fuzzy positions, e.g. 'less than 4'.

Xref subclasses

PublicationXref

Definition: An xref that defines a reference to a publication such as a book, journal article, web page, or software manual. The reference may or may not be in a database, although references to

PubMed are preferred when possible. The publication should make a direct reference to the instance it is attached to.

Comment: Publication xrefs should make use of PubMed IDs wherever possible. The DB property of an xref to an entry in PubMed should use the string “PubMed” and not “MEDLINE”.

Examples: PubMed:10234245

Properties:

The following properties may be used when the DB and ID fields cannot be used, such as when referencing a publication that is not in PubMed. The URL property should not be used to reference publications that can be uniquely referenced using a DB, ID pair. One reason for this is that it is expected that DB, ID pairs are more stable than URLs.

author - The authors of this publication, one per property value.

source - The source in which the reference was published, such as: a book title, or a journal title and volume and pages.

title - The title of the publication.

url - The URL at which the publication can be found, if it is available through the Web.

year - The year in which this publication was published.

RelationshipXref

Definition: An xref that defines a reference to an entity in an external resource that does not have the same biological identity as the referring entity.

Comment: There is currently no controlled vocabulary of relationship types for BioPAX, though one is needed and could be created in the future.

Examples: A link between a gene G in a BioPAX data collection, and the protein product P of that gene in an external database. This is not a unification xref because G and P are different biological entities (one is a gene and one is a protein). Another example is a relationship xref for a protein that refers to the Gene Ontology biological process, e.g. ‘immune response,’ that the protein is involved in.

Properties:

relationshipType - This property names the type of relationship between the BioPAX object linked from, and the external object linked to, such as 'gene of this protein', or 'protein with similar sequence'.

UnificationXref

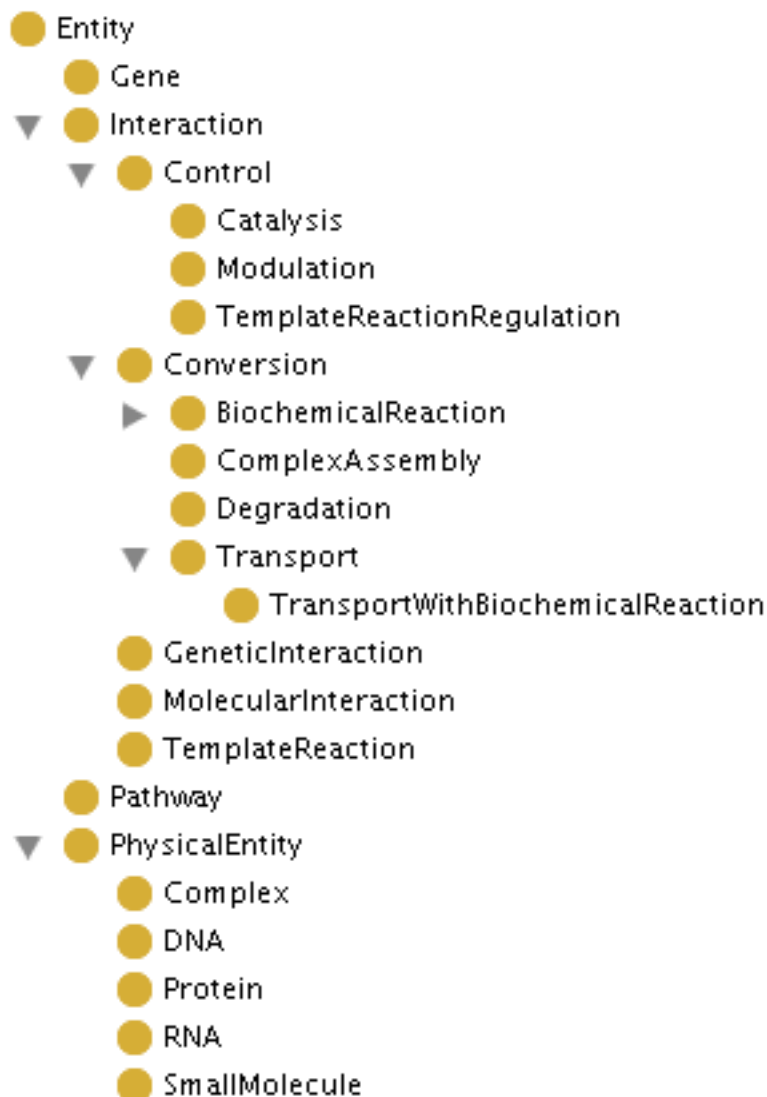
Definition: A unification xref defines a reference to an entity in an external resource that has the same biological identity as the referring entity¹⁶. For example, if one wished to link from a database record, C, describing a chemical compound in a BioPAX data collection to a record, C', describing the same chemical compound in an external database, one would use a unification xref since records C and C' describe the same biological identity. Generally, unification xrefs

should be used whenever possible, although there are cases where they might not be useful, such as application to application data exchange.

Comment: Unification xrefs in physical entities are essential for data integration, but are less important in interactions. This is because unification xrefs on the physical entities in an interaction can be used to compute the equivalence of two interactions of the same type. An xref in a protein pointing to a gene, e.g. in the LocusLink database¹⁷, would not be a unification xref since the two entities do not have the same biological identity (one is a protein, the other is a gene). Instead, this link should be captured as a relationship xref¹⁶. References to an external controlled vocabulary term within the OpenControlledVocabulary class should use a unification xref where possible (e.g. GO:0005737)

Examples: An xref in a protein instance pointing to an entry in the Swiss-Prot database, and an xref in an RNA instance pointing to the corresponding RNA sequence in the RefSeq database.

Summary of BioPAX Class Structure



- UtilityClass
 - BioSource
 - ChemicalStructure
 - ▼ ● ControlledVocabulary
 - CellularLocationVocabulary
 - CellVocabulary
 - EntityReferenceGroupVocabulary
 - EvidenceCodeVocabulary
 - ExperimentalFormVocabulary
 - InteractionVocabulary
 - PhenotypeVocabulary
 - SequenceLocationVocabulary
 - SequenceModificationVocabulary
 - TissueVocabulary
 - DeltaGPrime0
 - ▼ ● EntityFeature
 - BindingFeature
 - ModificationFeature
 - ▼ ● EntityReference
 - DnaReference
 - ProteinReference
 - RnaReference
 - SmallMoleculeReference
 - Evidence
 - ExperimentalForm
 - KPrime
 - ▶ ● PathwayStep
 - Provenance
 - Score
 - ▼ ● SequenceLocation
 - SequenceInterval
 - SequenceSite
 - Stoichiometry
 - ▼ ● Xref
 - PublicationXref
 - RelationshipXref
 - UnificationXref

3 Examples

A number of examples of pathways in the BioPAX format are available for download from the BioPAX homepage (<http://www.biopax.org/>) and from databases that support BioPAX.

4 Best Practices

While the BioPAX ontology imposes many logical constraints so that data encoded make sense for the use cases envisioned, some parts of the ontology have the potential for encoding data in multiple ways or there may be multiple options for treating data. This section recommends best practices in the use of the ontology for data exchange between groups. It supplements recommendations made in the class and property definitions provided above. It is expected that major data providers follow these recommendations to ensure compatibility of their data with other BioPAX data.

Users of BioPAX who are not exchanging data between groups, e.g. using BioPAX as an internal data model for their software, might find alternate representations to the ones recommended here more useful for their purposes.

Referencing External Objects

BioPAX is focused on representing interactions and pathways, but links to many different types of information. It is important to maintain these links, since Biological objects in external databases, such as proteins and small molecules, should be referenced via instances of the Xref class, and external controlled vocabulary terms, such as those defined by the Gene Ontology Consortium or the PSI-MI initiative, should be referenced via instances of the ControlledVocabulary class. BioPAX does not currently support a general mechanism to use RDF IDs to seamlessly point to external biological objects and controlled vocabulary terms on the semantic web.

Using xrefs

External references (Xrefs) relate elements within a BioPAX document to external data. Xrefs are more than just identifiers, as they contain the name of the data source the identifiers are part of and potentially version information as well. They exist to uniquely point to a record in an external data source (e.g. a bioinformatics database). For example, a pointer from a protein instance in BioPAX to a record in a database describing the protein would be established with an xref. Note: Xrefs are NOT related to RDF IDs.

Within any xref, database names (in the DB property) should be from a controlled vocabulary to avoid data integration problems that arise when different people use different spellings of database names. The PSI-MI controlled vocabulary includes a database name controlled vocabulary (see ‘Using external controlled vocabulary terms’ section for more information). If it is not possible to use this controlled vocabulary for a specific database, be careful to use the database name exactly as spelled on the database website (e.g. Use “Swiss-Prot” instead of swissprot, SWP or other frequent spellings). Also, suggest that the database name you want to use be added to the PSI-MI vocabulary. Similarly, the ID property should use the primary key of the target object, e.g. “P54352” instead of “HUMAN_P53”. Software should be able to use xref information to construct a web hyperlink to the database record being pointed to.

Xrefs to database accession numbers that contain version information should keep the version information separate from the identifier (ID), e.g. the accession number “CAA61361.2” should be stored as “id=CAA61361”, and “idVersion=2”. This is to enable computer software to easily identify the accession or the version without having to be aware of all possible ways of encoding the version in the accession number. If you are unsure how to encode the ID, think about how the encoded information could be used to build a web hyperlink to the referenced database record.

Importance of unification xrefs

Abundant use of unification xrefs, where possible, is highly recommended, especially in physicalEntity instances. These xrefs allow a user to understand that two independent instances from different BioPAX documents are actually the same entity (as long as they share one or more unification xrefs).

When exporting data from a database with primary keys, those keys should generally be encoded as unification xrefs. For example, if a database contains biochemical reactions with IDs for both the reactions and the small molecules that participate in those reactions, unification xrefs containing these IDs should appear in the corresponding BioPAX instances generated by the database. In general, the original data record from which an instance was generated should be pointed to via a unification xref. The exception to this rule occurs when the native class of the data is not completely synonymous with the BioPAX class to which it is mapped. In these cases, the resulting BioPAX instances should point back to the original data records via relationship xrefs.

Caution: Complications with unification xrefs can arise when the database that is being pointed to contains redundant information or contains more than one type of record. If a database contains redundant information, such as GenBank or Chemical Abstracts Service (CAS), it is possible to reference the same physical entity in the same database, but use IDs of different redundant records. In this case, unification xrefs can not be guaranteed to be useful in determining if two physical entities are the same across multiple BioPAX documents. More information about database record relationships will be required. Also, if a database contains different types of records, such as mRNA and protein records in GenBank or chemical structures with and without R groups in CAS, then it may be impossible to determine the type of record referenced, which may lead to errors from unification xrefs that point to molecules of a different type than the referencing physical entity. Care in creating unification xrefs should be taken when linking to these types of databases so that the link is unambiguous.

External database identifiers

Use of standard database names and identifiers is recommended. For instance, UniProt or RefSeq IDs should be used for proteins. A list of database names is available in the PSI-MI controlled vocabulary at <http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI&termId=MI%3A0444&termName=database%20citation>. Also, the MIRIAM resource maintains a list of database names, and additional information about each database, such as a regular expression that can be used to validate the ID (see <http://www.ebi.ac.uk/compneur-srv/miriam-main/>).

Using external controlled vocabulary terms

A number of properties in the BioPAX ontology reference the ControlledVocabulary class. Some of these are referred to as “mission-critical” because the information they provide is very important to most users of pathway data and to enable software to make simplifying assumptions about which vocabularies to expect. These are: CellularLocation, EvidenceCode, ExperimentalFormDescription, SequenceModificationVocabulary and SequenceRegionVocabulary. It is required that the following external controlled vocabularies (CVs) be used for these mission-critical properties:

CellularLocation: Gene Ontology Cellular Component

EvidenceCode: BioCyc Evidence Ontology, PSI-MI interaction detection CV (Note: the PSI-MI CV will likely be extended to contain all BioCyc evidence codes. When this occurs, the PSI-MI CV will be preferred.)

ExperimentalFormDescription: PSI-MI participant identification method (e.g. mass spectrometry), experimental role (e.g. bait, prey), experimental preparation (e.g. expression level) type CVs

SequenceModificationVocabulary: PSI-MI Feature Type CV, biological feature child terms

SequenceRegionVocabulary: Sequence Ontology

db property (in xref): PSI-MI Database Name CV, where possible

Note: evidenceCode allows multiple CV terms to be included. Because it is mission critical, at least one term should be from the above recommended CV.

Several other properties, cellType, interactionType, and tissue, also make use of the ControlledVocabulary class. These non-mission-critical properties serve simply to provide additional annotation. The following CVs are suggested for these properties:

cellType: <http://obo.sourceforge.net/cgi-bin/detail.cgi?cell>

interactionType: PSI-MI Interaction Type CV

tissue: BRENDA Tissue Ontology (BTO)

Note: The PSI-MI Level 2.5 CVs may be found at:

<http://psidev.cvs.sourceforge.net/psidev/psi/mi/rel25/data/>

Mission critical CVs that are relatively stable may be packaged with BioPAX as a convenience and are periodically updated with BioPAX releases, though developers should always consult the latest versions listed above for potential changes.

Cellular location

The location referred to by this property should be as specific as is known. If an interaction is known to occur in multiple locations, separate interactions must be created for each different location. Note: If a location is unknown then no term should be specified. Do not use the GO term for 'cellular component unknown' (GO:0008372). If the location of a participant in a complex is unspecified, it may be assumed to be the same location as that of the complex. In case of conflicting information, the location of the most outer layer of any nesting should be considered correct. Cellular location describes a specific location of a physical entity as it would

be used in e.g. a transport reaction. It does not describe all of the possible locations that the physical entity could possibly be in the cell, as would be listed in all known GO cellular component annotations for the protein.

Reusing utility class instances

Utility classes store structured bits of information in the context of the main ontology classes ('entity' and its subclasses). As such, they are not guaranteed to make sense out of the context of the classes they are used in. Utility classes should be reused carefully to avoid making improper statements out of context. For example, consider a `BiochemicalPathwayStep` instance that was used by multiple pathways. If new information became available for one of those pathways, addition of this additional information to the `BiochemicalPathwayStep` instance could invalidate it for all of the other pathways that refer to it. Due to these potential problems, it should not be assumed that utility class instances will be re-used in a BioPAX file. Software implementations must be aware of this if instance equality is important, so that equality statements are made based on all content of utility class instances.

Pathways and networks

In BioPAX, a pathway is defined using interactions and/or pathway instances. This provides sufficient flexibility to support two main representation conventions, the typical biochemical pathway, composed of a set of interactions, possibly ordered by `pathwaySteps` and possibly defined using sub-pathways, and the typical molecular interaction network, composed of a set of interactions not involving pathway steps or sub-pathways.

Black box pathways

The `pathwayComponent` property may be left empty, in which case the pathway would simply have a name and could be treated as a black box.

Pathway ordering

Pathway ordering is stored in the *pathwayOrder* property of the `Pathway` class. A pathway step should not be listed in the *nextStep* property of another `PathwayStep` if the intersection of the entities in the *participants* properties of their interactions is empty. Typically, at least one product of the conversion in each preceding `PathwayStep` should participate either as a *controller* or as a substrate to the conversion interaction of a `PathwayStep`. Note: The *nextStep* property is meant only to represent pathway topology, not order of events (see definition of the `PathwayStep` class in section 2 for more information). Holes in the pathway are allowed, for instance, if intermediate steps are not known.

Interaction networks

Often, molecular interaction datasets do not contain any notion of a pathway, but instead simply store a collection of binary or higher order interactions between molecules. For these datasets, instances of the `pathway` class are not necessary, though may be used to store sub-networks of the overall interaction network that are part of a pathway.

Control 'controller' and 'controlled' property conventions

Instances of Control can have multiple controller's and controlled's. Moreover, one Control instance can control another Control instance. The semantics of the use of these properties are as follows:

Multiple separate controls controlling a conversion means that they control in parallel (e.g. different enzymes catalyzing the same reaction). Generally, their effect on the rate of the reaction is cumulative.

A control with multiple controllers indicates a dependency between these controllers, typically meaning that both are required for the reaction to occur (e.g. a catalysis with an enzyme and a cofactor as controllers). Any further chaining of controls also implies dependency, for example allosteric inhibition of the aforementioned enzyme by a small molecule.

Here is a pseudo-BioPAX representation of the examples above:
rxn1 is a BiochemicalReaction

cat1 is a Catalysis
cat2 is a Catalysis

mod1 is a Modulation

enzyme1 is a Protein
enzyme2 is a Protein

cofactor1 is a SmallMolecule
drug1 is a SmallMolecule

cat1 has controlled rxn1
cat2 has controlled rxn1 (Both cat1 and cat2 can catalyze rxn1, independently)

cat1 has controller enzyme1

cat2 has controller enzyme2
cat2 has cofactor cofactor1 (both enzyme2 and cofactor1 is required for cat2 to occur)

mod1 has controlled cat2
mod1 has control-type INHIBITION_ALLOSTERIC
mod1 has controller drug1 (drug1 should NOT be present for cat2 to occur)

This structure is similar to disjunctive normal form (DNF) in Boolean logic. We could write this as: (enzyme1) OR (enzyme2 AND cofactor1 AND NOT drug1)

Conversion direction

Multiple places exist in BioPAX for providing information on the direction in which a conversion interaction proceeds. The *direction* property of the catalysis instance, if specified, should override all other sources of direction information. If the conversion is not catalyzed, or the *direction* property is empty, the *spontaneous* property of the conversion should be used as the source of direction information. If a conversion is spontaneous, then it will occur in the specified direction without any catalyst (although, in the cell, the reverse may happen by unknown processes). If no values for *direction* or *spontaneous* are specified, it may be possible to infer direction given the thermodynamic constants in the biochemical reaction, if specified and if assumptions about the conditions in the cell are made. It may be possible to infer direction using other computational techniques, such as flux-balance analysis¹⁸. The topology information from any PathwayStep instances in the pathway class should not be used for direction information, however, the stepDirection property of BiochemicalPathwayStep can be used to infer direction of a pathway step, in the context of a pathway (this is needed when the reaction in the step is reversible, but proceeds in one direction in a pathway context e.g. due to the law of mass action). Do not assume that the default direction of conversions is in either the LEFT-to-RIGHT or RIGHT-to-LEFT directions.

Degradation

Degradation spontaneous direction can only be L-R, NOT-SPONTANEOUS or unknown. It cannot be R-L. Also, any catalysis that references a degradation conversion cannot specify a R-L direction. Degradation usually contains nothing on the right side (we don't model degradation products). However, some cases involving complexes where part of the complex degrades, but the rest does not, like mdm2-p53 and ubiquitin ligases that target a partner for degradation can have right side participants.

Conventions for 'left' and 'right' properties of conversion

As stated above, substrates and products of a conversion may be placed in either the *left* or the *right* properties as these are not used to determine the direction of a conversion. However, in order to ease data integration, it is preferable that users adhere to the same conventions for the contents of these properties. We therefore recommend the following, in order of precedence:

1. If the conversion has an Enzyme Commission (EC) number or a Transport Commission (TC) number, store the participants in the LEFT and RIGHT properties such that they mirror the EC/TC reaction.
2. For complex assemblies, store the subunits in the LEFT property and the complex in the RIGHT property.
3. For transport instances, store the outermost participants (relative to the interior of the cell or organelle) in the LEFT property and the innermost participants in the RIGHT property.
4. If none of the above are applicable, store the participants from left-to-right in the order that the conversion occurs or is suspected to occur in the pathway.

stepDirection Property of BiochemicalPathwayStep

If 'direction' of the Catalysis instance contained in the step is "PHYSIOL-LEFT-TO-RIGHT", then stepDirection of BiochemicalPathwayStep is blank (unknown, unspecified) or LEFT-TO-

RIGHT. If stepDirection of BiochemicalPathwayStep is not empty then 'direction' of the Catalysis instance is either blank, "REVERSIBLE" or "PHYSIOL-LEFT-TO-RIGHT".

Technical note: OWL and RDF Conventions

A typical set of BioPAX data consists of many instances of various BioPAX classes. Each of these instances must be given an RDF ID that is unique within the document to be a valid OWL/RDF document. These IDs are used to reference instances from other parts of the OWL document. When combined with a globally unique document namespace, these IDs form a URI that can provide a globally unique identifier for each BioPAX instance.

RDF ID

In an OWL document, such as BioPAX, each instance of a class must have an RDF ID. This comes from the Resource Descriptor Framework standard (<http://www.w3.org/RDF/>). These IDs must be unique and are used to reference instances within a document. An RDF ID exists within a namespace, which can be explicitly appended before the RDF ID. If not explicit, the RDF ID exists in the default namespace of the document. Like anchors in HTML, a pointer to an RDF ID defined elsewhere in the document is denoted with a hash mark (“#”) in front of the RDF ID.

Example

```
<protein rdf:ID="protein76">  
  <XREF rdf:resource="#xref1146"/>  
</protein>
```

It is recommended that RDF IDs do not encode any semantics and be composed of the class name followed by a unique positive integer (e.g. “protein76”) or some other naming convention that guarantees unique names within the file. Some applications that use OWL, such as Protégé, and some examples of OWL from the main OWL website, use human readable names for the RDF IDs. As long as these names are unique, a BioPAX document will be valid, but the use of human readable names as RDF IDs might encourage people to rely on information stored in them and is thus not recommended. RDF IDs may not persist after certain data processing operations, such as integrating data from two separate BioPAX files.

Please note that in the Protégé tool, the RDF ID of an instance is referred to as its Name. This should not be confused with the BioPAX *name* property, which is meant to provide the human readable name for biological entities (Figure 1). Protégé can be configured to display the value of the *name* property (or another field value) instead of the RDF ID. Use the Display Slot pull-down menu in the Individuals tab to select the value to display.

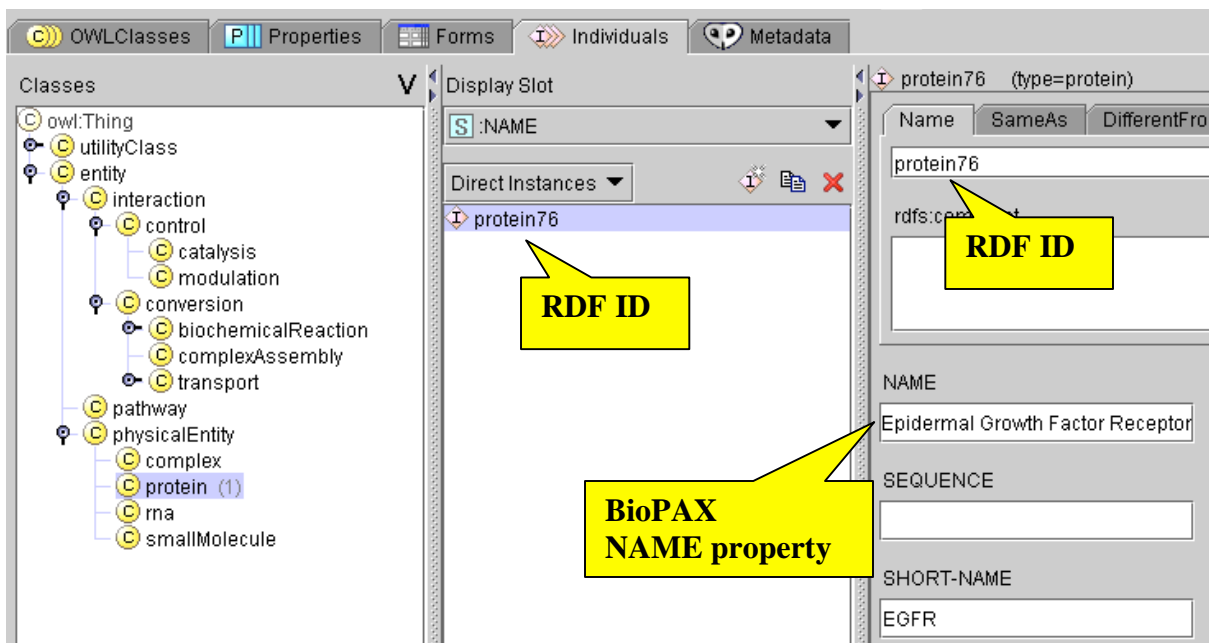


Figure 1: The difference between name and RDF ID shown in Protégé.

Document namespace

OWL XML documents require a default namespace. The creator of the BioPAX document should create a namespace and encode it in the BioPAX document. The namespace and the RDF ID may be used together to reference instances in a document from an external document (explicit use of namespace). This reference mechanism is part of the basis of the planned Semantic Web (<http://www.w3.org/2001/sw/>). If a BioPAX document is going to be on the Semantic Web, it should have a unique namespace. Since there is no namespace naming authority, it is not possible to guarantee unique namespaces across the internet, but following these recommendations will reduce the chances of naming collisions.

Technically, any string without spaces is allowed (see [namespace rules](#)) as a namespace. Operationally, a URL (or more generally a URI) should be used. This does not have to be a ‘real’ URL that resolves to a web page, but it should be related to the organization of the creator and a registered domain name owned by the organization is useful to include e.g. “<http://biocyc.org/ontology/biopax/#>”.

Use of the `xmlns` and `xml:base` attributes to specify the namespace for any BioPAX documents created is recommended. The BioPAX ontology definition should be imported and the BioPAX namespace should be defined using the ‘bp’ string (if it does not conflict with other existing namespaces called ‘bp’) e.g. `xmlns:bp=http://www.biopax.org/release/biopax-level3.owl`, so that elements in the file appear like this: `<bp:pathway></bp:pathway>`.

A typical header of an OWL XML document that uses the BioPAX ontology will look like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<rdf:RDF
  xmlns="http://www.myorganization.org/ontology#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xs="http://www.w3.org/2001/XMLSchema#"
  xmlns:bp="http://www.biopax.org/release/biopax-level3.owl#"
  xml:base="http://www.myorganization.org/ontology"
>
<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.biopax.org/release/biopax-level3.owl"/>
</owl:Ontology>
```

Where “<http://www.myorganization.org/ontology#>” defines the namespace for this document.

OWL XML documents that mix BioPAX definitions with those from other ontologies or extend BioPAX will have different ways of using namespaces, but that is outside the scope of this document and will likely not be valid BioPAX Level 3 documents. It is good practice to specify the character encoding in the XML header, in this case UTF-8. If you have international characters in your BioPAX document, be sure to specify the correct character encoding. See <http://www.w3.org/International/O-charset.html> for more information.

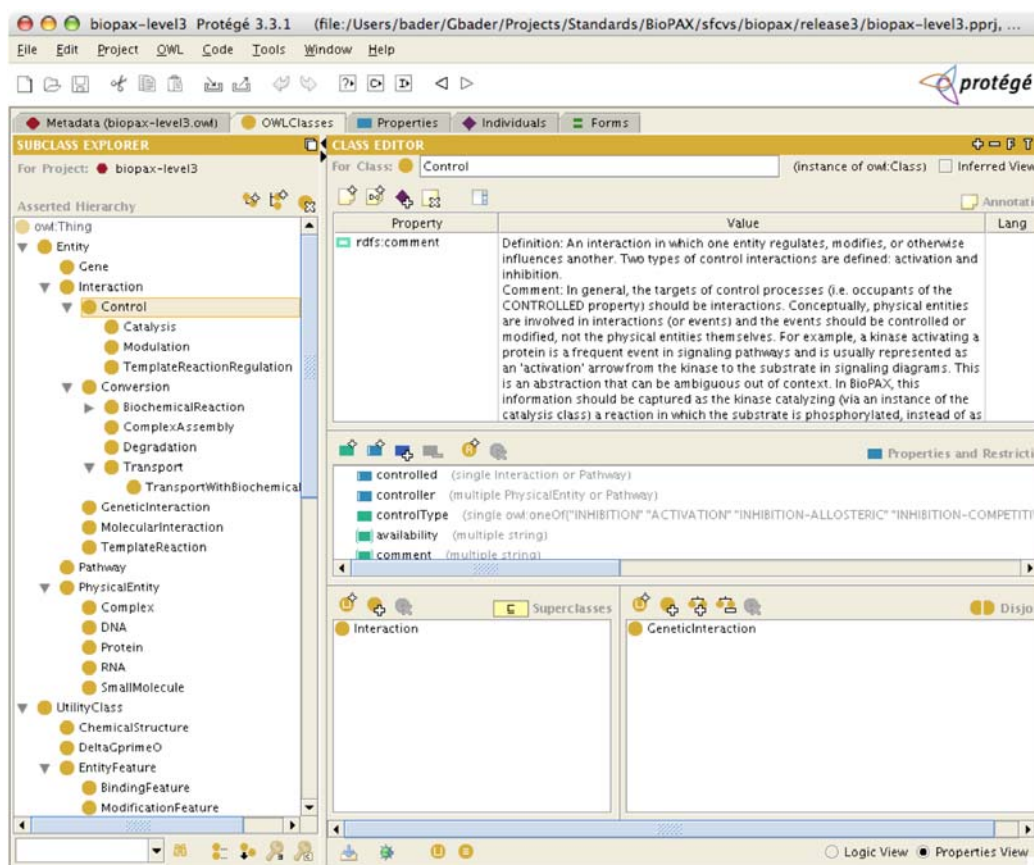
5 HOW-TO

Creating a knowledge-base using BioPAX and Protégé

Protégé is an open-source ontology and knowledge-base editor from Stanford University. It can be used to view and edit the BioPAX ontology and to create a database of instances of BioPAX classes. Download Protégé from <http://protege.stanford.edu/> Downloading the current stable release and not the beta release is recommended. Make sure your Protégé version comes with OWL support. Follow the instructions for installing Protégé. These instructions were written for Protégé 3.3.1.

To import from a local copy of the BioPAX OWL file:

- 1) Load the BioPAX OWL file via the “File → Open...” menu item. In the resulting dialog box, select “OWL Files” and browse to the BioPAX OWL file on the local computer disk drive by clicking on the + symbol next to “OWL file name” and press “OK”. Protégé will load BioPAX.
- 2) Upon loading, the BioPAX ontology will be visible OWLClasses Tab. Ensure the properties view is selected, instead of the logic view.
- 3) Use the Individuals tab to create instances.



Note: This method of importing BioPAX into Protégé does not prevent inadvertently made changes to the imported BioPAX classes; changing the ontology is not recommended if the instance data are meant to be shared.

Protégé can be used as a full-fledged customizable database and data entry system, although it requires programming effort. For example, Reactome (<http://www.reactome.org>) uses Protégé as its backend system. If used this way, it may be desirable to modify the BioPAX ontology and create inverse properties for convenience. These properties should be removed in shared data files in order to make them compliant with the BioPAX standard.

Viewing Instances Graphically

Instances can be graphically viewed with a number of Protégé plugins that ship with the ‘Full’ protégé download. For instance, the Ontoviz plugin enables a highly customized view of instances in an OWL file.

6 Use Case Outlines

These use-cases were considered during the design of BioPAX. Other use-cases may be suggested via the biopax-discuss@biopax.org mailing list.

Data Sharing Between Databases

One of the primary intended functions of the BioPAX format is to facilitate data exchange between existing biological pathway databases. In order for this to happen, databases must develop the ability to write-to and read-from the BioPAX format. Typically, this will require the creation of in-house software. While a number of freely available software packages may help make this task easier (e.g. Jena, an open source Java API for RDF; see <http://jena.sourceforge.net/index.html> or the Protégé OWL API; see <http://protege.stanford.edu/plugins/owl/>), development of data translation software may nonetheless require a fair amount of programming time for each individual database. This can be significantly reduced using the PaxTools Java library for reading, writing and validating BioPAX files.

The typical data transaction, i.e. passing a set of data from one database to another, will consist of a number of steps. These steps will vary depending on the particular situation, but in general they should consist of the following:

- 1) Convert a set of data into the BioPAX format. This step involves mapping the native data model to the BioPAX data model (i.e. the BioPAX ontology) and then creating a BioPAX OWL file that contains instances of the mapped classes. This step will almost always require developing software to perform the mapping.
- 2) Transfer the BioPAX file. There are many mechanisms by which this could be accomplished, e.g. the data provider could make the file available for download from an FTP or HTTP server.
- 3) Convert the BioPAX file into the native format of the receiving database (the reverse of step 1). Again, this will likely require new software to perform the data conversion.
- 4) Merge data sets and remove redundancies. Often, many instances in a BioPAX file may already exist in the target database (Note: these are only detectable if the redundant instances share one or more unification x-refs or if entire instances are compared). These instances should be merged with the existing data (if they contain additional information not present in the database) or removed from the data set being imported (if not) to prevent redundant entries from being created. Also, any pointers to such instances must be redirected to the existing database objects.

As more datasets become available in the BioPAX format, software utilities will be developed (by members of the BioPAX group and others) to ease data sharing. For example, a utility to integrate the data from two different BioPAX files would be useful. With such a utility, users could integrate new BioPAX data with their own by first outputting their data into BioPAX format, then running the utility to combine it with the new data, then translating the combined

data set back into their own format. Thus, the need for system-specific data integration software (step 4 above) would be reduced.

BioPAX as a Knowledge-Base (KB) Model

The BioPAX ontology is readily usable as the data model for a pathway knowledge-base (KB) using a tool like Protégé (<http://protege.stanford.edu>). Building a new KB with the BioPAX ontology would save time and resources since it would eliminate the need to create a data schema from scratch and it would reduce the translation requirement for exporting and importing data to/from the BioPAX format (some custom semantic mapping and ID mapping might still be required to import data from another database).

Of course, some users may wish to extend the BioPAX ontology to suit their own needs. For example, many KBs use “inverse properties” – properties that are the reciprocal of other relationship properties – in order to speed up queries and facilitate browsing. Since such properties provide redundant information, they were left out of the BioPAX ontology. See the HOW-TO section for more information on creating a BioPAX KB. Note that instances adhering to an altered BioPAX ontology are not compatible with the official BioPAX standard unless converted back to standard BioPAX.

Pathway Data Warehouse

An initial motivation for creating the BioPAX standard was that it was seen as a logical first step toward creating a central public repository for biological pathway data, a resource strongly desired by many members of the pathway community. If many databases provide access to their data in the BioPAX format, it should be relatively simple to aggregate this data in a central repository, like Pathway Commons (www.pathwaycommons.org).

Pathway Analysis Software

Another intended function of BioPAX is to speed development time of software that makes use of pathway data. Currently, in order for pathway software to access pathway data from multiple sources it must either be programmed to interpret each different format, or the data from each source must be translated into a format that the software supports. This can require significant development time and as a consequence most pathway analyses are run on only a few datasets, limiting utility.

The presence of a standard format and object model for pathway data should alleviate this problem. With the lower barrier to data access, pathway software will be easier to develop and apply. Also, additional software that might not be practical without an agreed upon standard, e.g. a sophisticated pathway visualization tool, may be more likely to be developed if BioPAX becomes widely adopted.

Pathway Analysis Software Example: Molecular profiling analysis

Genomics and proteomics technologies, such as gene expression microarrays and mass spectrometers, are being used to generate large datasets of molecules present at a specific place and time in an organism (molecular profiling), among other types of data. Molecular profiling

experiments are often compared across two or more conditions (e.g. normal tissue and cancerous tissue). The result of this comparison is often a large list of genes that are differentially present in the tissue of interest. It is interesting and useful to analyze these lists of genes in the context of pathways. For instance, one could look for pathways that are statistically over-represented in the list of differentially expressed genes. The result is a list of pathways that are active or inactive in the condition of interest compared to a control. The list of pathways is often much shorter than the list of input genes, thus is easier to comprehend. BioPAX documents describing pathways could be supported by tools that perform pathway-based analysis.

Visualizing Pathway Diagrams

Pathway diagrams are useful for examining pathway data. A number of formats are available for these images, but only few available viewing tools link components in the image to underlying data. A mapping of BioPAX to a symbol library for pathway diagrams (such as Kohn maps - <http://discover.nci.nih.gov/kohnk/symbols.html>) could be the basis for a general BioPAX pathway diagram tool. An example of a tool that supports visualizing BioPAX pathways is PATIKAWeb (<http://web.patika.org>).

Pathway Modeling

Mathematical modeling to understand the dynamics of a pathway system is a frequent use of pathway information. Qualitative modeling requires information about components in the pathway and their connections, as well as some qualitative knowledge of rates (e.g. fast, slow) and concentrations of the components (e.g. high, medium, low). Quantitative modeling additionally requires such things as measured rate constants, stoichiometry and initial concentrations in order to quantitatively predict pathway behavior. Many tools are available for this type of modeling, and the SBML (<http://sbml.org>) and CellML (<http://www.cellml.org>) standards are available to describe the models, which many tools support. While BioPAX does not contain enough information to describe a pathway model as well as SBML and CellML, there are two envisioned use cases:

Using BioPAX as metadata for SBML and CellML

SBML and CellML, as model representation languages, focus on representing the structure, parameters and mathematical description of a pathway model. BioPAX focuses on molecule and interaction classification schemes and database cross-referencing for pathway components. BioPAX and SBML or CellML could be linked together when a user wants both a full model description and information about types of pathway components and database links. A hybrid XML document containing BioPAX and SBML or CellML elements that are tied together using the CellML metadata standards could be created that fills this need.

Pathway analysis using logical inference

One advantage of representing BioPAX pathway data in OWL format is the availability of logical inference tools that support OWL. These tools are useful for analyzing pathways. For example, given a metabolic network model for an organism in BioPAX format, a known minimal nutrient media for that organism and the set of compounds essential for growth under one set of living conditions, then a transitive closure computation of the minimal nutrient set can be used to verify if the metabolic network model of the organism is sufficient to explain growth. If any

essential compound is not reachable through the network from the minimal nutrient list, then the network model is incomplete.

7 Glossary

Some of the following definitions may be specific to BioPAX.

Biological pathway: A pathway is a series of molecular interactions and reactions (or other biological relationships), often forming a network. For molecular pathways, the start and end points are often defined by observation of a detectable phenotype after stimulation or perturbation, such as observing gene expression after stimulating the cell with a peptide growth hormone.

Class: Used in knowledge representation to represent a category of things. A specific member of a class is called an instance or individual.

Data exchange format: Any data format, usually electronic, used to exchange data.

Instance: An particular member of a class. Known as ‘individual’ in OWL.

Ontology: A system for describing knowledge, a conceptualization of a domain of interest usually made up of any or all of the following: concepts (classes), relations, attributes, constraints, objects, values. <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>

OWL: Web ontology language, a proposed W3C standard, is an extension of RDF to support ontologies. It provides semantics for classes and subclasses, instances, and relationships. <http://www.w3.org/TR/owl-features/>

Property: A ‘field’ or ‘member’ of a data structure. Known as a ‘slot’ in many knowledge representation systems.

Protégé: Protégé ontology and knowledge base editor. A software tool to build an ontology and manage instances of classes defined in that ontology. <http://protege.stanford.edu/>

RDF: Resource Description Framework, a proposed W3C standard, allows description of basic relationships between objects (subject-predicate-object semantics). OWL is built on top of RDF. <http://www.w3.org/TR/rdf-primer/>

Appendix A: Design Principles

Flexible: Biological pathway data are organized and represented in various ways depending on the type of data and its intended use. BioPAX must support the most frequently used representations to be widely accepted. Of course, there is a trade-off that must be considered: increased flexibility may increase data integration overhead. For example, the issue of semantic mapping between different representation styles must be dealt with when users wish to integrate BioPAX data sets that use different representations. Therefore, BioPAX should strike a reasonable balance between flexibility and rigidity by allowing multiple preferred representations and providing best practice recommendations to encourage consistent data representation.

Extensible: Biological pathway data are available in various forms and at varied levels of detail. BioPAX aims to initially support the most frequently used types of pathway data and levels of detail and to progressively broaden support for additional pathway data types and finer detail through a leveled approach. The class structure of BioPAX was designed to be extensible for this reason. Many parts of the BioPAX ontology, such as internal controlled vocabularies and many of the intermediate level classes, will be extended in future BioPAX levels. All efforts will be made to keep future levels backwards compatible.

Encapsulation: Pathway data depends on many primary databases of physical entities (e.g. proteins, small molecules, etc.). Many pathway data sets reference physical entities using database identifiers. Because of the varied nature of the physical entity databases, resolving these identifiers in a general way can be difficult, especially for the naïve user. Frequently used data about the physical entities (e.g. sequence for proteins, structure for small molecules) is optionally present (encapsulated) in the BioPAX format for convenience.

Compatible: BioPAX uses existing standards for encoding biological pathway information to avoid “re-inventing the wheel”. Specifically, pointers to the Gene Ontology (GO), and instances of Chemical Markup Language (CML) and the SMILES format are used in various properties in the ontology. Also, compatibility with other pathway standards, such as SBML, CellML, and PSI-MI has influenced the design of many BioPAX features.

Computable: BioPAX stores data in a format that supports many different types of computational analysis. Values are strongly typed and the class structure is clearly defined. A wide range of computational tasks, from simple reading and parsing of a BioPAX file to logical inference based on the data, are supported. The OWL version of the BioPAX ontology is written in the OWL-DL sublanguage and is thus intended to be interpretable by description logic software such as RACER (<http://www.sts.tu-harburg.de/~r.f.moeller/racer/>). However, please, see Appendix C: BioPAX Non-Conformance with OWL Semantics regarding our use of OWL.

Appendix B: Level and Version Numbers

BioPAX level numbers indicate the relative scope of the ontology. BioPAX Level 1 focuses on metabolic pathway data; Level 2 expand this scope to include molecular binding interactions; Level 3 adds support for signal transduction pathways, gene regulatory networks and genetic interactions. BioPAX level numbers are always whole numbers (e.g. Level 1, version 1.0).

In addition to the level numbers, BioPAX version numbers indicate the relative stage of development of each level. Version numbers are a composite of two individual integers: the major version number and the minor version number separated by a decimal point to form the composite version number (e.g. Level 1, version 1.1). The major version number appears before the first decimal point and is only incremented when an update is likely to affect existing data. Releases in which the major version is 0 are early draft releases of their respective levels (e.g. Level 1, version 0.5).

The minor version appears after the first decimal point and is incremented when an update is unlikely to affect data that conforms to the prior version. Odd minor version numbers indicate beta versions, while even minor version numbers indicate release versions.

For example, the first non-draft release of every level is version 1.0 (major version 1, minor version 0). If this version would need to be updated, the first beta version of the update would be called version 1.1. When no further revisions were needed, a release version of the update would be created called version 1.2 if the revisions would not affect data that complied with version 1.0, or version 2.0 if they would.

All versions of BioPAX are available in the following directory on the BioPAX website:
<http://www.biopax.org/Downloads/>

The most recent major versions of each level of BioPAX are always available in this directory:
<http://www.biopax.org/release/>

Appendix C: BioPAX Non-Conformance with OWL Semantics

BioPAX Level 3 use of OWL does not fully conform to the OWL semantics. There is discussion in the community about the biological semantics of specific classes. BioPAX Level 3 does not fully respect OWL semantics because, until recently, we didn't understand them¹⁹. We are using OWL similar to a UML style class hierarchy definition language with a standard XML serialization. OWL follows an "open world assumption", but BioPAX Level 2 in some cases assumes a "closed world", similar to most data schema definition languages. In particular:

- There are a number of cases where open world semantics conflict with the intended semantics of BioPAX entries. For example, in the case of metabolic reactions it is often (but not always) intended that the list of participants in the reactions is considered to be complete. However we do not add closing axioms to make our OWL representation say so. On the other hand it would not be correct to say that the current BioPAX operates under closed-world semantics. One counterexample would be the representation of reactions where the catalyst is not known. Another would be assuming that a BioPAX file contains all reactions of a certain class - if a particular reaction is not found in a BioPAX file, you should not assume that it does not exist.
 - BioPAX often uses domain and range where the meaning we intended should be expressed using restrictions.
 - Aspects of the meaning of some terms, such as Pathway, are embedded in the comments for the class, rather than being made explicit in the definitions of the classes.
 - Cardinality restrictions are often used as a means to suggest which properties are required or optional, rather than being considered definitional.
- BioPAX Level 3 assumes that there are no user-defined classes and that exchange files only contain instances.
 - In order to ensure correct parsing of BioPAX files, and to support future levels, it is recommended that OWL-parsing tools, such as Jena, are used instead of non OWL-aware XML parsing tools such as SAX and DOM.

Appendix D: Change log

This section contains the technical change log for recent BioPAX OWL files. A complete change log is available from the BioPAX SourceForge CVS system at <http://biopax.cvs.sourceforge.net/biopax>

Specifically:

<http://biopax.cvs.sourceforge.net/biopax/Paxtools/src/org/biopax/paxtools/model/biopax-level3.owl?view=log>

<http://biopax.cvs.sourceforge.net/biopax/biopax/release3/biopax-level3.owl?view=log>

BioPAX Level 3, Release candidate 3 (version 0.92) – Released March 14, 2008

- Fixed bug in componentStoichiometry, range changed from PhysicalEntity to Stoichiometry
- name property's domain is generalized to include all Entities (previously interactions were excluded)
- evidence property's domain is generalized to include all Entities (previously Gene was excluded)
- ReferenceEntity is renamed to EntityReference to better reflect semantics. All subclasses renamed in a similar fashion
- ReferenceEntityGroupVocabulary renamed to EntityReferenceGroupVocabulary
- Removed SequenceLocationGroup, made all sequence locations groupable.
- Introduced restrictions on subclasses of SequenceLocations members to make sure only groupings of all sites or all intervals are valid.
- Removed ExternalReferenceUtilityClass
- Renamed DataSource to Provenance
- Removed property sourceName, added regular name property instead
- Removed name and xref from Score
- Created functional property scoreSource(Source,Provenance)
- Renamed properties pH to ph, pMG to pMg
- Renamed class DeltaGprimeO to DeltaGPrime0
- Restriction on Degradation Right = 0 was removed to accommodate (A-B)-->A(degraded) + B type of representations.
- Added change log to documentation (request from Frank Schacherer/BioBase)
- Added warning to documentation about physicalEntity semantics change from Level 2 to Level 3 (request from Frank Schacherer/BioBase)
- Added experimentalFeature property to ExperimentalForm class to handle physicalEntity features that are purely experimental (and increase compatibility with PSI-MI 2.5)
- Changed property experimentalFormType to experimentalFormDescription to better reflect types of controlled vocabularies it refers to.
- PaxTools implementation of Level 3 used to test the entire ontology (which found many of the issues fixed in this release).

BioPAX Level 3, Release candidate 2 (version 0.91) – Released December 22, 2007

This release candidate is a major update based on suggestions on mailing list and from SRI meeting, October 2007, including new implementations of gene regulation, genetic interaction and degradation.

- Changed all classes to CamelCase and properties to mixedCase (request by Matt Halstead and others)
- Change CONTROLLER property domain to include pathways. Request from NCI/Nature PID
- Fixed errors associated with multiple use of the same property by different classes
- PHYSICAL-ENTITY needs gene in experimental form, but not in stoichiometry - fixed by creating an experimental-form-type property.
- Changed binds-to property to symmetric
- Renamed covalentFeature to modificationFeature
- Renamed nonCovalentBinding to bindingFeature
- Changed openControlledVocabulary to controlledVocabulary
- Moved ReferenceEntity back to utility class - to prevent it from the ability to participate in interactions
- Changed modified-at property name to feature and not-modified-at property name to not-feature
- Added 'reversible' to step-direction in biochemicalPathwayStep (request by Suzanne Paley/BioCyc)
- Changed modification feature to use modifications
- Changed binding feature to use sequence region vocabulary instead of sequence feature type.
- Changed genetic interaction phenotype to use phenotypeVocabulary class, rather than phenotype, which allows use of PATO and other phenotype CVs.
- Renamed physicalInteraction to molecularInteraction (request from Ken Fukuda/INOH)
- Renamed nonCovalentFeature to nonCovalentBinding (request from Paul Thomas/PANTHER)
- Updated documentation on not-modified-at property to make it clearer that the default is 'don't know' (request from Nigam Shah/NCBO and Peter Karp/BioCyc)
- Changed sequenceLocation.location-type to sequenceLocationVocabulary - bug fix from Andrea Splendiani
- Renamed sequenceLocationVocabulary to sequenceRegionVocabulary
- Renamed entityFeatureVocabulary to sequenceFeatureVocabulary
- Removed disjoint statements between complex and small molecule and its siblings, same for reference entities (request from Oliver Ruebenacker)
- Removed disjoint statements between interaction children, except geneticInteraction and all siblings.
- Removed disjoint statements from control and conversion children.
- Updated small molecule definition (request from Oliver Ruebenacker)
- Removed unused properties: gene-member-region, gene-product, reference-entity-component
- Removed all disjoints in the utility class section - they were incomplete, so we should evaluate them again closer to release. (if we choose to make this permanent, this saves ~15kb of file size and will make the autogenerated documentation much easier to read)
- Moved shared physicalEntity properties up to physicalEntity: binds-to, cellular-location, modified-at, not-modified-at
- Modified covalentFeature comment - spotted by Oliver Ruebenacker
- Added new subtypes for openControlledVocabulary
- Removed IN-COMPLEX - no reverse properties now and added documentation for several properties. All classes and properties have descriptions now.

- Significant property and class documentation update
- Gene regulation proposal implementation:
 - Renamed templateInteraction to templateReaction
 - Deleted expression class
 - Renamed expressionRegulation to templateReactionRegulation
- Added templateReaction:
 - template: RNA, DNA
 - product: dna, rna, protein
 - regulatory-element: e.g. promoter, RBS, etc. (type DNA, RNA)
- TemplateReactionRegulation:
 - Controller: physicalEntity
 - Controlled: templateReaction
- Implementation of degradation:
 - New degradation class, child of conversion
- Implementation of genetic interaction:
 - Added gene class, child of entity (not a physical entity). This is only used for genetic interactions.
 - Change confidence class to score, so it can be used in genetic interaction
 - Re-implementation of phenotype as a class, so that multiple ways of defining phenotype can be used and so it is more extensible to future OWL representation of phenotypes.

BioPAX Level 3, Release candidate 1 (version 0.9) – Released August 30, 2007

Proposals for States, Generics, Genetic Interactions and Small Fixes were added to BioPAX Level 2. Gene regulation proposal was still under discussion and wasn't included.

References

- 1 Andreas D. Baxevanis and B. F. Francis Ouellette, *Bioinformatics : a practical guide to the analysis of genes and proteins*, 2nd ed. (Wiley-Interscience, New York, 2001); Bruce Alberts, *Molecular biology of the cell*, 4th ed. (Garland Science, New York, 2002).
- 2 L. Stein, **417** (6885), 119 (2002).
- 3 H. Hermjakob, L. Montecchi-Palazzi, G. Bader et al., *Nat Biotechnol* **22** (2), 177 (2004).
- 4 P.D. Karp, M. Riley, M. Saier et al., *Nucleic Acids Res.* **30** (1), 56 (2002); C. J. Krieger, P. Zhang, L. A. Mueller et al., *Nucleic Acids Res* **32 Database issue**, D438 (2004).
- 5 G.D. Bader, D. Betel, and C.W. Hogue, *Nucleic Acids Res.* **31** (1), 248 (2003).
- 6 R. Overbeek, N. Larsen, G.D. Pusch et al., **28** (1), 123 (2000).
- 7 E. Demir, O. Babur, U. Dogrusoz et al., **18** (7), 996 (2002).
- 8 G. Joshi-Tope, M. Gillespie, I. Vastrik et al., *Nucleic Acids Res* **33 Database Issue**, D428 (2005).
- 9 C. Lemer, E. Antezana, F. Couche et al., *Nucleic Acids Res* **32 Database issue**, D443 (2004).
- 10 M. Kanehisa, S. Goto, S. Kawashima et al., *Nucleic Acids Res* **32 Database issue**, D277 (2004).
- 11 H. Mi, B. Lazareva-Ulitsky, R. Loo et al., *Nucleic Acids Res* **33** (Database issue), D284 (2005).
- 12 M. Hucka, A. Finney, H. M. Sauro et al., *Bioinformatics* **19** (4), 524 (2003).
- 13 P. Murray-Rust and H. S. Rzepa, *J Chem Inf Comput Sci* **43** (3), 757 (2003).
- 14 D. Weininger, **28**, 31 (1988).
- 15 The_Gene_Ontology_Consortium, **25** (1), 25 (2000).
- 16 P. D. Karp, *Trends Biotechnol* **14** (8), 273 (1996).
- 17 D.L. Wheeler, D.M. Church, S. Federhen et al., *Nucleic Acids Res.* **31** (1), 28 (2003).
- 18 J. S. Edwards, M. Covert, and B. Palsson, *Environ Microbiol* **4** (3), 133 (2002).
- 19 A. Ruttenberg, J.A. Rees, and J.S. Luciano, presented at the OWL Experiences and Directions, Galway, Ireland, 2005 (unpublished).