CURATOR GUIDE FOR PATHWAY/GENOME DATABASES

USING THE PATHWAY TOOLS SOFTWARE

VERSION 22.0

RON CASPI, CAROL FULCHER, INGRID KESELER,
MARKUS KRUMMENACKER, SUZANNE PALEY, AND PETER D. KARP

SRI INTERNATIONAL
333 RAVENSWOOD AVE.
MENLO PARK, CA 94025

ptools-support@ai.sri.com
MARCH 2018

Previous contributors: Martha Arnaud, John Ingraham, Cynthia Krieger

Table of Contents

1. INTRODUCTION

This curator's guide contains information for curators of Pathway/Genome Databases (PGDBs) with an emphasis on the EcoCyc [4] and MetaCyc [3] databases.

This guide addresses issues regarding PGDB conventions, literature search and review, PGDB data entry, editing, and maintenance. Since the roles of curators may vary, only parts of this guide may be relevant to any particular reader. Some sections of this guide may be specific to the curation of a particular PGDB and not be applicable to other PGDBs.

Another important source of information for curators is the Pathway Tools User's Guide document, which is accessible from the desktop version of Pathway Tools under the Help menu.

1.1. DEFINITIONS

Pathway/Genome DataBase (PGDB): A database that describes the genome of an organism (replicon(s), genes, and genome sequence), the product of each gene, the metabolites used by the organism, the biochemical reaction(s) catalyzed by the organism's enzymes, and the organization of reactions into pathways. A PGDB can also describe the genetic network of an organism: its promoters, operons, transcription factors, and transcription-factor binding sites. A PGDB is a type of MOD (Model Organism Database).

EcoCyc: The first PGDB, created for the bacterium Escherichia coli K12 strain MG1655. This is the only PGDB that was created without the help of PathoLogic, since it predates PathoLogic. EcoCyc is the most comprehensive organism-specific PGDB currently available.

MetaCyc: MetaCyc is a multi-organism PGDB that contains metabolic data from many different organisms. The goal of MetaCyc is to catalog the universe of metabolism by storing a representative sample of each experimentally elucidated pathway. The pathways in MetaCyc are used by the PathoLogic component of the Pathway Tools program to predict the pathway complement of a particular organism, which is modeled within a separate PGDB for that organism. The majority of the information in MetaCyc is derived from the biomedical literature.

The MetaCyc family of databases: This is a term for all PGDBs that were generated by Pathway Tools using MetaCyc as a template. There are many thousands of such databases, some created

by SRI and some created by other groups. They all share a common schema, and are thus compatible, providing a useful platform for comparative analyses.

The BioCyc Database Collection: The collection of PGDBs available at the biocyc.org website. This collection includes all of the databases created by SRI, as well as some additional databases submitted by external groups.

Tier levels: The BioCyc databases are divided into three tiers, based on their quality. Tier 1 databases have received at least one year of manual literature-based curation (some, like EcoCyc and MetaCyc, received decades of curation). Tier 2 databases received a small amount of manual curation, while Tier 3 databases contain only computationally predicted information.

1.2. OVERVIEW OF PGDB CONTENT

Most PGDBs provide a unique combination of genome information and the metabolism of one organism (a rare exception are multi-organism PGDBs, such as MetaCyc and PlantCyc, that do not contain full genome information). The fundamental building blocks of the genomic part of the database are the replicons, while the fundamental building blocks of the metabolic part are chemical compounds. Starting with these building blocks, PGDBs contain many different types of objects that describe complex entities and processes.

Examples of objects associated with the genomic part of the database are genes, transcription units, and proteins.

Examples of objects associated with the metabolic part of the database are compounds, reactions, and pathways.

This version of the curator's guide does not cover the curation of all possible objects, but rather focuses on compounds, reactions, pathways, genes and enzymes.

Future versions will be expanded to cover additional types of objects.

1.3. SOURCES OF INFORMATION

It is up to a curation team to decide which sources of information are to be used for the curation of their PGDB. For curation of tier 1 PGDBs at SRI (MetaCyc, EcoCyc, HumanCyc, YeastCyc and BsubCyc) the information has to be published in a peer-reviewed article or a

book. The vast majority of information in most of the databases originates from the scientific literature, although open-source published PhD theses are also acceptable.

The most comprehensive free database for biomedical literature is PubMed. It provides a quick and easy starting point for researching information about genes, enzymes, and metabolic pathways. The PubMed ID (PMID) can be used for the automatic retrieval of the reference information into Pathway Tools, and provides an instant web link from the PGDB to PubMed.

It is often possible to find papers not cover by PubMed by simple Google or GoogleScholar searches.

A very comprehensive source for information about enzymes is the UniProt website, which provides not only specific information about proteins but also links to other databases and literature references. Note that UniProt contains a combination of annotated SwissProt entries and non-annotated TrEMBL entries. The curated entries are of much higher value. More information about enzyme, including excellent references, is available at the BRENDA website.

For information about chemical compounds, ChEBI provides accurate structures and a chemical ontology for many compounds. Reliable structures for compounds missing from ChEBI may be found at ChemSpider or KEGG COMPOUND. Other databases, such as PubChem, are less reliable, and should be used with caution.

The definitive resource for the Enzyme Commission (EC) system is found at ExplorEnz. However, this information is also available within MetaCyc where it is usually up-to-date.

Taxonomy information is found at the NCBI taxonomy homepage.

1.4. NAMING ISSUES

Common names and synonyms: It is important to follow strict guidelines when naming objects in the database. Every object in Pathway Tools has a common name and optional synonyms. It is very important to maintain consistency in the common names of related objects. It is also important to provide synonyms to maximize the chances that a user will find the object when searching for it. For example, for D glucose 6-phosphate we also provide the synonym D-glucose-6-P. Note that when performing searches the software removes punctuation and lowercases all names, so there is no need to enter synonyms that differ only in punctuation or capitalization.

When entering synonyms, one should be precise and not enter incorrect or misleading synonyms. For example, do not enter the synonym "D-glucose" for the compounds α-D-glucose and β-D-glucose. D-glucose is the specific name of the parent class of these two compounds, and does not map directly to either of the two anomers.

Names in the literature: Most scientists are not nomenclature experts, and past experience has shown that names used in the literature can be inappropriate. Never assume that a name is correct simply because it appeared in a paper. Try to use common sense, and consult with your peers when unsure.

Internal Hyperlinks: It is very useful to refer (within comments) to objects in the database by an internal hyperlink instead of typing the name. This way, if the name changes, a single change to the common name of that object will alter the way references to that object appear throughout the database. Using internal hyperlinks is also beneficial for the users, and minimizes the risk of typos.

2. GENERAL GUIDELINES FOR PGDB CURATION

Each type of object in a PGDB is edited using a different editor. Some types of curated material are common to many objects (e.g. references to the literature) while others are object-specific. This section provides guidelines for common topics, whereas later sections (under Section 3 Curation) discuss the curation of specific database objects.

2.1. SUMMARIES

Text stored in the Comment slot of an object will be seen by the public under the heading "Summary:". We call this text "mini-review summaries" or "summaries" for short.

Please try to minimize redundancy between information within summaries and information in other parts of the PGDB, whenever possible. When there are dedicated structured fields for storing information (e.g. enzyme kinetics, cofactors or inhibitors) the data should be entered into those fields rather than in the summary. Keep in mind that, unlike the dedicated fields, the information in the summary field is only meant for a human reader and is not understood by the software. For example, if an enzyme is inhibited by calcium, do not write "The enzyme is inhibited by calcium" in the summary. Instead, enter this information into an inhibitor field. In practice, it is probably not possible to completely eliminate redundancy, because some of the important biology that curators will want to write in summaries will be redundant with other PGDB data. When a curator does choose to put redundant information in a summary, the information should be important, and the summary should try to provide additional information beyond what is present in structured database fields.

2.1.1. WRITING STYLE GUIDELINES FOR SUMMARIES

Summaries should be written in full sentences. Use multiple paragraphs within summaries where the extra white space adds clarity and separates ideas. Keep in mind that due to a limitation of the software, a single line break is ignored. If you want to start a new paragraph, you must hit the Enter key twice.

Embed citations within summaries in a way that makes the link between the citation and the text it supports clear. Other than general commentary, most of the text of a summary should consist of an assertion followed by a citation, then another assertion followed by a citation.

Pathway summaries often benefit from providing background material (e.g. discussion of a compound that is degraded or synthesized, or discussion of different pathway variants that exist). When providing such background, include it in the first paragraph under the heading "General Background" (in bold text). Write the pathway-specific part of the summary in a following paragraph, under the heading "About This Pathway".

Abbreviations should be defined at their first use within a summary, in parentheses.

Always use American English. If you copy and paste some text from a paper, make sure you modify British English spelling to the American counterpart (e.g. change sulphur to sulfur). This convention maintains consistency and helps users know which variation to search for.

Avoid stating "recently" or "currently" as those statements do not hold for long – they may be read by users at a much later time than they were written. Use instead "in 2014" or "as of 2014…".

2.1.2. FORMATTING TEXT WITHIN SUMMARIES

Pathway Tools supports a subset of HTML tags for encoding special characters and formatting within summaries and names. Following is a list of tags that are accepted.

For example, to encode the name α-D-glucose, enter the characters: "α-D-glucose".

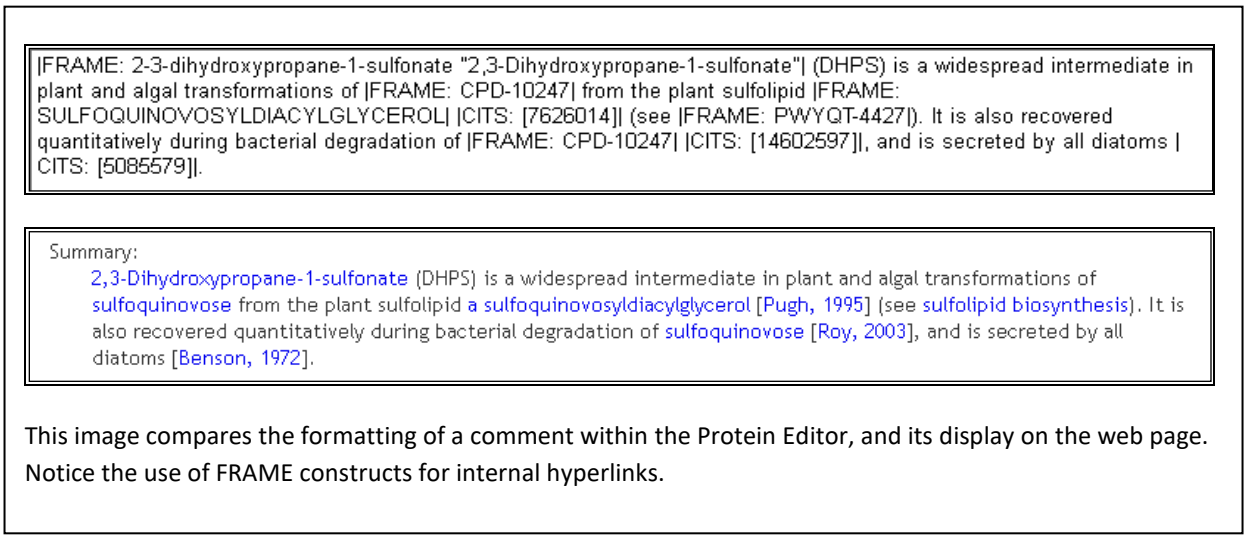
- Examples of common Greek letters used in Biology
  - α: α
  - β: β
  - γ: γ
  - δ: δ
  - Δ: Δ
  - ω: ω
  - Ω: Ω
  - μ (micro): μ

- Text
  - italicized text: text
  - bold text: text
  - underlined text: <u>text</u>
  - superscript: ^{text}
  - subscript: _{text}
- Special characters
  - Angstrom (˚A): Å
  - Degrees (°): °
  - Tilde (~): ˜
  - left arrow (←): ←
  - right arrow (→): →
  - double-sided arrow (↔): ↔
  - & &
  - Mdash (—): —

The HTML tags <P> (new paragraph) and
 (hard line break) are removed from the displayed text and not observed. Thus, to force a new paragraph, hit the return key twice instead of using <P>.

2.1.3. INTERNAL HYPERLINKS

It is very useful to refer (within summaries) to objects in the database by an internal hyperlink instead of typing the name. This way, if the name is found to be incorrect, or (as is often the case for bacterial taxonomy) changes, a single change to the common name of that object will alter the way references to that object appear in summaries throughout the database. In addition, it ensures uniformity across all entries in the database, and is beneficial for the users, allowing them to easily navigate to objects mentioned in the comment.



This image compares the formatting of a comment within the Protein Editor, and its display on the web page. Notice the use of FRAME constructs for internal hyperlinks.

To use internal hyperlinks, first visit the object you would like to refer to, so it is placed in the history list. Then open the editor, place the cursor where you want the hyperlink, and click on the FRAME button. This opens a window with the content of the history list. Click on the object that you want to link to, and a reference in the format |FRAME: FRAME-ID|will be inserted into the text, where FRAME-ID is the frame ID in the object you are linking to. When the history list becomes long and it is difficult to find objects in it, you can reset it by selecting Tools → History → Clear.

The use of the FRAME button and/or the history list is not mandatory – it is there only for convenience. If you want to type the frame ID instead of using the history list, it is often convenient to visit the page and print the frame ID to the Lisp buffer by using right-click → Show → Show frame name. Once the frame ID appears in the Lisp buffer, you can copy it to the operating system's clipboard (e.g. using control-C) and paste it into the editor.

When you use a hyperlink the linked object will appear in the text with its common name (see below for an exception regarding reactions). If you would like to modify the text that appears in the comment (for example, the link is the first word in a sentence and thus should be capitalized), you can add the desired display text in the following format:

|FRAME: FRAME-ID "display text"|.

Linking to reactions: when entering a hyperlink using a reaction frame-ID, the reaction equation will show up in the text unless you specify a display text.

2.1.4. SAY IT IN YOUR OWN WORDS

Avoid word-for-word duplication from papers (that is plagiarism). If you find the need to do so, enclose the text (no more than one to three sentences) within quotes and cite the source. We must hold ourselves to a high standard in this regard – with the tens of thousands of eyes trained on our databases, plagiarism is likely to be detected quickly.

2.1.5. CITATION GUIDELINES

Citations should be used within summaries to cite the source of the information just conveyed. Embed citations in a way that makes the link between the citation and the text it supports clear. Do not lump all the citations at the end of the summary, because it is not clear which assertion(s) a given citation pertains to. However, if relevant reviews are available, they should be grouped together at the end of the summary, such as: "Reviews: [Smith95,Jones98]."

Citations can also be entered independently of summaries, in the citations boxes found in the editors. This manner of adding citations is less desirable, since it is not clear which statements (if any) are supported by these citations. Use the citation boxes only if you feel that including a citation is useful, but you do not find any better place to store it.

2.1.6. HISTORY NOTES AND INTERNAL COMMENTS

Curators can record explanation and justification of edits to a PGDB object in a history note, created using the command Right-Click → Notes → Add to History. Example commentary could describe the reasons for changing a gene function, a chemical structure, or a pathway definition. History note contents are public, and will be displayed with the date they were created and the username of the creating curator.

Should you want to record information that will not be seen by the public but that will be visible in the Navigator to curators, put that information in the Comment-Internal slot using the Frame Editor.

2.2. USING CLASSIFICATION (ONTOLOGY) SYSTEMS

PGDBs contain several classification systems to which individual objects can be assigned. Such systems exist for chemical compounds, reactions (the EC system), pathways, and genes (the MultiFun system). Classification of objects facilitates searching and drill-down browsing by users.

In some cases it is appropriate to assign an object to multiple classes. Specifically, always classify superpathways under both the superpathways class and a suitable other class describing the biological significance of the pathway.

When updating an object, please consider whether its assignment within a classification system should be revised.

2.3. SAVING CHANGES

If changes have been made to a database, an asterisk will appear to the left of the database name at the top of the navigator window, for example, *MetaCyc. Changes are not saved automatically. It is important to remember to click the "Save DB" button often to save changes.

If something goes wrong during the save process, and you are unable to save your changes, use command File → Checkpoint Current DB Updates to File. This will save all changes to a temporary text file. You should then restart Pathway Tools and execute the command File → Restore Updates from Checkpoint File, then Save DB. If you are still not able to save your changes, you should consult an SRI programmer.

2.4. Evidence Codes

PGDBs include an evidence ontology that is designed to encode information about why we believe certain assertions in a PGDB, the sources of those assertions, and the degree of confidence scientists hold in those assertions.

A detailed description of the evidence ontology can be found in [1], although some evidence codes have been added or revised since that publication. The current evidence ontology can be browsed within a PGDB by running the GKB Editor on the PGDB class hierarchy rooted at class Evidence. An HTML version of the evidence ontology is available at URL

http://brg.ai.sri.com/evidence-ontology/downloads/evidence.html

Curators can assign evidence codes by clicking on the "Evidence Code" button within most Pathway Tools Editors. For example, within the protein editor, there is an Evidence Code button below the Enzyme Activity box. This button allows the curator to assign an evidence code; a citation to the source of the evidence should also be added when available (the citation is

optional since some evidence codes such as EV-AS-NAS (Non-traceable author statement) are used when no citation is available). Whenever an evidence code has been assigned, a new "Evidence Code" button will be drawn to allow assignment of an additional evidence code. It is proper to assign multiple evidence codes if multiple types of evidence support a given conclusion. In rare cases, a curator may feel confident about an entered fact, even though there is no direct literature reference to support it. For example, if the curator identified a protein in the databases by performing Blast searches using a sequence reported in the original paper, or perhaps chromosomal location of neighboring genes reported in the paper. In such cases, you may use the evidence code Inferred by Curator.

When assigning evidence codes, consider the following issues.

Pathways. Assign one of the sub-codes of EV-EXP (inferred from experiment) to a pathway if some experimental evidence supports its existence. Code EV-COMP (inferred from computation) or its sub-codes should be used for pathways whose presence is inferred computationally, such as by the PathoLogic program. Since MetaCyc contains only experimentally elucidated pathways, all pathways in MetaCyc are assigned an EV-EXP code.

Proteins. Evidence codes should be assigned to a protein to define the evidence supporting the function of the protein. For example, was the function of the protein elucidated using sequence analysis, or using experimental methods; if the latter, what class of method was used?

Enzymatic Activities. Several evidence codes specific to enzymes capture evidence from experimental methods that are specific to elucidating enzyme activities, such as EV-EXP-IMP-Reaction-Enhanced (gene is isolated and over-expressed, and increased accumulation of reaction product is observed). These evidence codes are stored within the Enzymatic-Reaction frames.

3. CURATION OF SPECIFIC OBJECT TYPES

This section provides guidelines for curation of specific database objects.

3.1. COMPOUNDS

This section discusses the curation of compounds.

3.1.1. BACKGROUND: ORGANIZATION OF THE COMPOUND CLASS HIERARCHY

Compounds are organized in a hierarchical class ontology for several reasons, two of which are enabling the specification of groups of compounds that are interchangeable as enzyme substrates in generic reactions, and allowing user navigation and retrieval of sets of compounds that share functional groups or metabolic purposes.

Classifying chemical compounds in an ontology is not simple, since there are several types of applicable classification systems, all of which provide some utility. For example, compounds can be classified based on their functional groups (e.g. All-Amines, All-Carboxy-Acids, All-Amino-Acids, All-Carbohydrates) or on their metabolic roles (e.g. Coenzymes, Hormones, Vitamins, Secondary-Metabolites). In some cases compounds appear to lack a distinguishing feature, making their classification particularly challenging. The classification hierarchies are not fully developed and are constantly being expanded, refined and rearranged. One of the best sources for compound ontologies is the ChEBI database and curators are encouraged to consult that ontology.

Newly created compounds are created by default under the top class “a small molecule“ (frame-ID Compounds), unless another class is explicitly specified. Many compounds in MetaCyc are still classified under that class.

The common name of most classes starts with “a” or “an,” e.g. “an alcohol”. An exception is lower level classes that holds related instances that differ, for example, by their stereochemistry (for example, the class D-glucose).

When creating new compound classes, the frame IDs should have every individual word capitalized, as in All-Carboxy-Acids.

3.1.2. BROAD SUBSTRATE SPECIFICITY

Compound classes allow a curator to represent broad substrate specificities in reaction equations. By using a compound class as a substrate in a reaction equation, you are stating that all instances of that class are acceptable substrates for the enzyme catalyzing the reaction. An example for a generic reaction is the following:

a 1-lysophosphatidylcholine + H_2O = a fatty acid + L-1-glycero-3-phosphocholine

The class “a 1-lysophosphatidylcholine” on the left side stands for a group of compounds that have a variable-length fatty acid tail. On the right side, the fatty acids class stands for the corresponding hydrolysis products (in contrast, H_2O and L-1-glycero-3-phosphocholine are specific compounds, represented as instances).

When using classes in a reaction, the curator has to make an informed decision about how much can be safely assumed about the likely extent of specificity in a given reaction equation.

Sometimes an enzyme with broad substrate specificity is known to not accept some of the members of a compound class. Rather than cluttering up the compound ontology with classes that attempt to exclude just those few exceptions, the proper procedure in these cases is to attach the enzyme to the generic reaction. If the system then shows instantiated reactions on that enzyme page that are known not to be catalyzed by the enzyme, it is possible to exclude those specific reactions by right-clicking on each undesired reaction, selecting Edit → Detach Enzyme(s), and selecting the enzyme.

3.1.3. COMPOUNDS ENTRY

Summary of information to be collected for chemical compounds:

- Chemical name and synonym(s)
- Parent class(es)
- Chemical structure
- Links to other databases
- Summary (optional)
- Citation(s)

For SRI curators: MetaCyc is considered the authority for compounds and reactions. Entering a new compound or altering an existing one should always be performed in MetaCyc. If you need to use the compound in another database, export it to that database after completing its curation in MetaCyc. See Section 4, Coordinating curation between MetaCyc and other Tier 1 PGDBs, for more information.

Names

- Most scientists are not nomenclature experts, and past experience has shown that names used in the literature are often incorrect. Rather than just copy the name from the literature, try to find the compound in ChEBI, where names are usually correct. If the compound is not found in ChEBI, try to find a similar compound in the database and name the new compound in a compatible manner. It is possible to find the systematic names of compounds by opening them in the Marvin structure editor and selecting "Naming" from the "Calculations" menu, but these names are not always practical.

- Compound names should not be capitalized except where uppercase characters are strictly required (e.g., use "L-tryptophan," not "L-Tryptophan").

- Since all compounds in our databases are programmatically protonated to pH of 7.3 and the chemical structures are of the conjugated bases, the common name of acids should be that of the conjugated base (e.g. acetate). Add the name of the acid (e.g. acetic acid) as a synonym.

- Special considerations in naming compounds: It is highly recommended to use HTML formatting for special characters in compound names. For example, instead of "->" use "→" (right arrow). While this makes no difference in the desktop version, the HTML characters look much better on the web server.

- The following characters cause breaks in the SVG image generation code and should never be used in names: **<->** (use ↔ instead) and **&** (use & instead).

- Polymeric compounds: Many reactions describe the addition or subtraction of a monomeric unit from a polymer. For example, EC 2.4.1.49:

  $[(1\rightarrow4)\text{-}\beta\text{-D-glucosyl}]_n$ + phosphate = $[(1\rightarrow4)\text{-}\beta\text{-D-glucosyl}]_{n-1}$ + α-D-glucose 1-phosphate

We keep only one compound frame for the polymeric compounds, so when specifying the reaction, the same compound appears on both sides. However, we can assign several special names to compounds, representing forms comprising n, (n-1), and (n+1) monomers. When specifying the reaction, it is possible to choose which name is shown on either side of the reaction (more about it in the reaction curation section).

Parent classes (compound ontology)

- This information is often not simple to obtain. If the compound has been classified under the ChEBI ontology, it is a lot easier. If the ChEBI parent classes are missing from MetaCyc, create those as well. Often there is a need to create several parent classes before the new compound can fit into the MetaCyc compound ontology.

- Become familiar with the compound ontology. It is a lot easier to find a suitable parent class when you are aware of its existence.

Chemical Structures

- The most reliable source for structures is ChEBI, followed by the original literature. Other databases often contain erroneous structures.

- The editor used for chemical structure editing is Marvin. It is a powerful tool, and it is highly recommended that curators learn how to use it well, including customization of the tool bars. Marvin is in transition from a Java-based applet to a JavaScript-based tool. Currently the JS tool (called Marvin JS) is not as powerful as the applet. However, the applet only runs on older versions of browsers due to security restrictions in current versions. An older Firefox version is the recommended platform for running the older Marvin applet.

- If a structure is available in ChEBI, you can retrieve it automatically (after entering the ChEBI link) by right clicking the compound name and selecting Edit → Import compound structure from ChEBI. However, it is often necessary to modify it after the import, since ChEBI does not have the same guidelines as MetaCyc (see below).

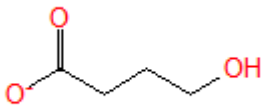
- Always make sure that the structure is protonated for pH 7.3. If you are not sure about the correct protonation, you can find out in Marvin by selecting Calculations → Protonation → Major Microspecies.

- You can ensure that your structure will look good by selecting the Marvin command Structure → Clean 2D → Clean in 2D. This command will produce straighter lines and fix distorted rings. However, it may also modify the structure in a way that would necessitate rearranging it after the cleanup.

- It is often easier to start with the structure of an existing compound than to create a new structure from scratch. To copy the structure of compound X to compound Y, start by right clicking on compound Y and selecting Show → Show frame name, which prints the frame ID in the lisp window. Copy the frame ID of compound Y to the operating system's clipboard.

Next open compound X in Marvin and paste the frame ID of compound Y instead of that of compound X (in the field next to the Submit button). Now make your changes and submit.

- Keep in mind that related structures should be drawn similarly, so when the structures are shown in reactions and pathways they are consistent and make it easy to follow the chemical transformations. This is often achieved by following strict conventions. For example, **a linear molecule should be drawn with carbon 1 on the left** (e.g. 4-hydroxybutanoate should be drawn with the carboxylate on the left and the hydroxyl group on the right). For a carboxylate, draw the carbonyl group pointing up. **A simple aromatic ring should be drawn so that the ring carbons are shown in a clockwise manner with carbon 1 at the top**.



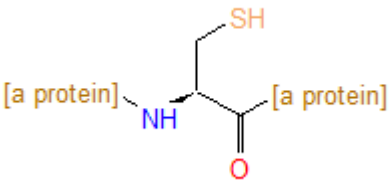
4-hydroxybutanoate. Note that carbon 4, with the hydroxyl group, is shown as the 4th carbon from the left.

- Marvin custom templates: Certain structural groups, such as coenzyme A, are rather complex. In order to avoid having to draw them from scratch, it is possible to add them as custom templates in Marvin. Once such a group is stored as a template, adding it to a structure involves one click of the mouse. To add a template to Marvin, follow this procedure (we will use the COA-GROUP frame as an example): Open the frame COA-GROUP in Marvin. Verify that you can see the template toolbar at the bottom of the screen. If you don't see it, select View → Toolbars → Advanced Templates. Select the whole structure in the Marvin main window and drag it to the template tool bar at the bottom. A small icon with the compound structure will be added to the tool bar. Right-click the icon and choose "properties", and add the abbreviation CoA. The template is ready.

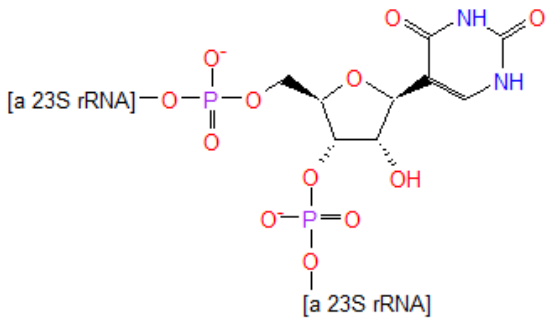
- Using R atoms: many compound classes can have structures associated with them, where the variability in the structure among the different members of the class is represented by an R atom. To enter an R atom, open the compound in Marvin and enter the structure using regular atoms. Now click on the "periodic table and more" icon (the top icon in the right side bar), click on the Advanced tab, and click on the Alias button. Type R in the value field and click on the atom you want to modify. If the structure contains several different variable groups, they should be represented by using R1, R2

etc. Marvin has specific buttons for specifying these, and there is no need to use the Alias button.

- Using frame names as compound atoms: we often use a frame ID as a pseudo-atom in Marvin. For example, an amino acid within a protein can have “Protein” as a pseudo-atom, as in the example below. To do so, first find the frame ID of the compound you want to use as the pseudo-atom (e.g. |Proteins|). Open the compound in Marvin and enter the structure using regular atoms. Now click on the “periodic table and more” icon (the top icon in the right side bar), click on the Advanced tab, and click on the Alias button. Paste the frame ID in the value field. You must include the vertical bars if the frame ID contains lowercase characters. Now click on the atoms you want to modify. They will change to the frame ID. However, when you submit the structure to Pathway Tools, the atoms will show up with the common name rather than the frame ID.



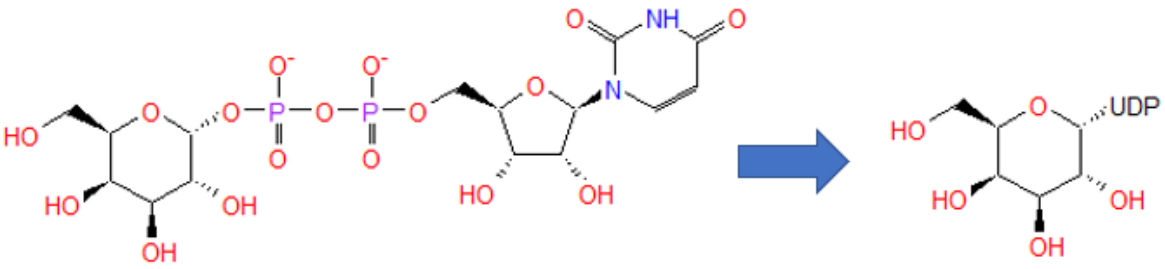
a [protein]-L-cysteine



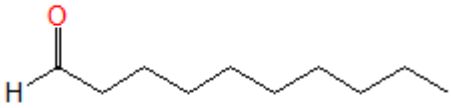
a pseudouridine in 23S rRNA

- Superatoms. When large groups are attached to a molecule to activate it (e.g. coenzyme A, or nucleotide phosphates), it is often advantageous to be able to collapse the structure of that group to a single atom (= superatom). When these structures are collapsed it is easier for a user to focus on the more important part of the molecule and observe the changes in reactions and pathways. In addition, in the desktop version of Pathway Tools, clicking on a collapsed superatom expands it to show the full structure. Superatoms are defined by creating a frame for them under the Superatoms class. To modify a compound to use the superatom feature, it is necessary to add the superatom’s frame ID as a value in the superatoms slot. Currently, this can be done only using the frame editor. For example, to shrink the structure of the UDP group in

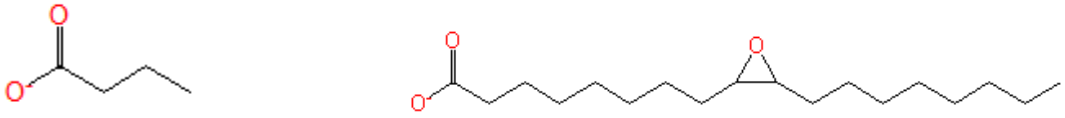
compound UDP-α-D-galactose, you would add the value UDP-GROUP in the superatom slot. The full and collapsed structures for this compound are shown below.



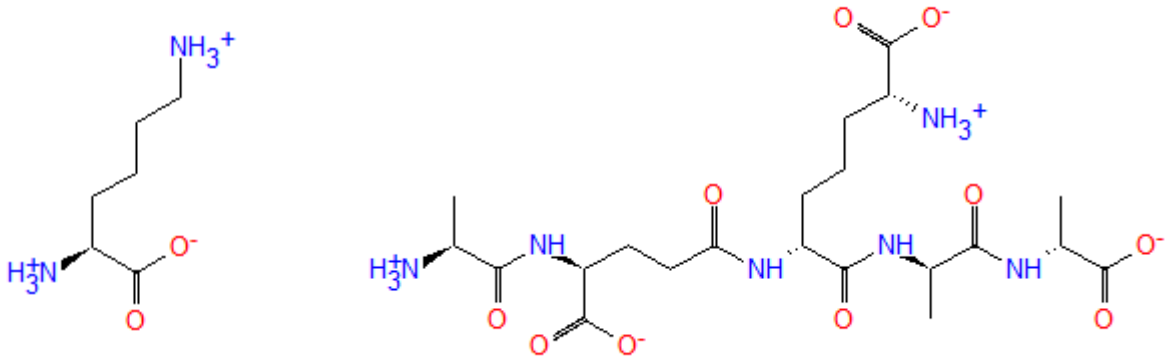
3.1.4. EXAMPLES FOR CHEMICAL STRUCTURES



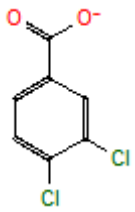
An aldehyde (1-decanal). Note the keto group is on the left, pointing up. In many cases it is useful to draw the hydrogen atom to indicate that it is an aldehyde.



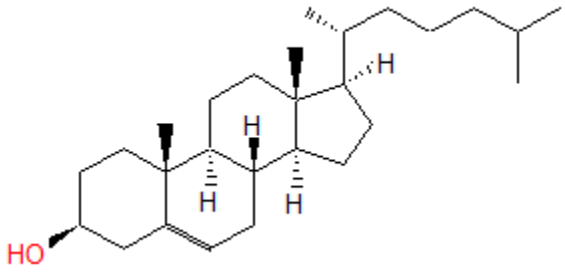
Carboxylic acids (*n*-butanoate and 9,10-epoxystearate. Note the carboxylate group is on the left, with the carbonyl group pointing up. The oxygen on the left is deprotonated.



An amino acid (L-lysine) and a pentapeptide. Amino acids are drawn differently than other carboxylic acids, with the amino group on the left and the carboxy group on the right (the way they appear in peptides). This makes it easy to copy and paste amino acid structures in order to combine them into peptides.



A modified benzene ring (3,4-dichlorobenzoate). Carbon 1 is always at the top, and the other carbons are shown in a clockwise manner.



A steroid (cholesterol). Note that the stereochemistry is depicted by adding hydrogen atoms where necessary. Do not use directional bonds within rings.

3.1.5. THE SCHEMA? SLOT

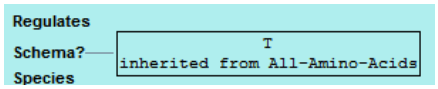
Compound classes are the building stones of the compound schema, and upon the creation of a new PGDB they are imported into the new database. In most cases this is desired, but there are exceptions. For example, let's look at the compound "a [SoxY protein]-L-cysteine".

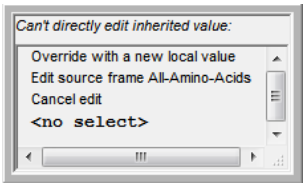
In general, when a compound describes a part of a protein structure, the compound is created as a class rather than an instance, because the protein can belong to different species and is thus a generic entity. In this example, the compound class describes an L-cysteine residue within a generic SoxY protein. Since many organisms do not have such a protein, it is undesirable to have this class propagated into every PGDB, regardless of whether it exists in that organism or not.

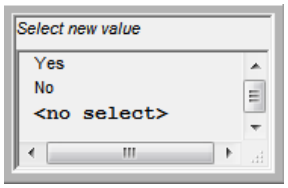
Whether a compound class is considered part of the schema is determined by the value attached to the "Schema?" Slot. The value in this slot is inherited from the parent class, and is usually set to YES by default. When creating a compound class that should not become part of the schema, that value should be changed manually to NIL.

Currently this can be be done only in the Frame Editor. Open the compound class in the Frame Editor, highlight the value attached to the Schema? Slot, and select Value -> Edit (or ^e).

If the value is inherited from the parent class, you will need to first select "Override with a new local value", followed by selecting No from the next popup.







3.2. REACTIONS

This section discusses the curation of reactions.

3.2.1. REACTION CURATION

Summary of information collected for reactions:

- Reaction equation
- Conversion type

- EC number
- Reaction name (when no EC number exists). Spontaneous reaction?
- Comment
- Citation(s)
- Reaction Directionality
- Spontaneous?

Reaction equation

- The reaction equation should be written in a direction that is compatible with the EC system. For example, reactions in the 1.1.1 sub-subclass are always written with the oxidized form of the cosubstrate (e.g. NAD+) on the left side and the reduced form (e.g. NADH + H+) on the right side. If you do not know what is the appropriate direction, look at similar reactions within the same sub-subclass.

- A correct reaction is both atom-balanced and charge-balanced. If you need to add protons to balance for hydrogen atoms, make sure that the charge is balanced. If not, you must have gotten something wrong.

- If the reaction cannot be balanced even though all of its components have structures (e.g. if it involves a polymeric reactant), make sure to check the button “this reaction cannot be balanced”.

- If the reaction cannot be balanced because some of its components are not known, check the button “this reaction cannot be balanced”, and in addition classify it as an “Unknown conversions” (see Conversion type below).

- If a reaction involves a generic entry such as “DNA” or “a protein”, you must use special holder classes that are stored under a class called “a nonspecific substrate” (frame-ID Holder-Class). These classes currently include “a protein” (General-Protein-Substrate), “a nonspecific small molecule” (Compounds-Holder-Class), “a generic polynucleotide substrate” (Polynucleotide-Holder), “DNA” (DNA-Holder) and “RNA” (RNA-Holder). NEVER use the classes with similar or identical common names that are the top classes that contain hundreds and thousands of subclasses and instances (such as |Proteins|) as a reaction substrate. To ensure the reaction editor uses the correct class, use the frame ID when entering these holder classes in the reaction editor.

Conversion type

- The most common conversion type is “Chemical Reactions” and this type is selected by default. However, there are multiple other types, such as redox half reactions, electron transfer reactions, transport reactions, composite reactions etc. Make sure that you select the correct type.
- Redox half reactions and electron transfer reactions are discussed below.
- Composite reactions are reactions that are a summary of at least two sub-reactions. If you specify a composite reaction, you need to specify (by frame IDs) its sub-reactions.
- Transport reactions must have at least two different locations for different reactants.
- When the reactions stands for a chemical transition that has not been completely defined yet, select “Unknown conversions”. The reaction will be depicted in pathway diagrams by three parallel arrows.

EC-numbers

MetaCyc contains four types of EC numbers. The following list explains the differences among the different types.

1. Formal EC numbers. Formal EC numbers contain only numerical digits (e.g. 3.2.1.45). These numbers are fully defined by the Enzyme Commission and can be found in the ExplorEnz database (www.enzyme-database.org). Reactions that are assigned these EC numbers can be marked as *official*, indicating the reaction is identical to that specified in the EC entry, or *non-official*, indicating that while it is known to be catalyzed by the enzyme defined in the EC entry (and by the same active site), it includes alternative substrates that are not part of the reaction(s) specified by the EC.
2. Temporary EC numbers. Temporary EC numbers contain numerical digits in the first three fields, but alphabetic characters in the fourth field (e.g. 2.1.1.ba). These numbers are used by the Enzyme Commission for new entries that are being drafted, and are replaced at the end of the process by a formal number. They should not be cited, as they are short-lived.
3. M-numbers. M-numbers contain numerical digits in all four fields, but the number in the fourth field is preceded by a capital M (e.g. 3.2.1.M7). These numbers are proprietary to MetaCyc and are assigned within MetaCyc to enzymes that do not have a formal EC number and are not in the process of being classified by the Enzyme Commission. Ideally, these numbers will be replaced by formal EC numbers once processed by the Enzyme

Commission. They are similar to N-numbers and B-numbers, which are found in the Uniprot and BRENDA databases, respectively.

4. Partial EC numbers. Partial EC numbers contain a hyphen character in one or more fields. These numbers should be used to indicate that the function of an enzyme is not well defined. For example, if sequence analysis suggests that an enzyme is a methyltransferase, yet the identity of the substrate is not known, it should be assigned the partial EC number 2.1.1.-. Ideally, this type of numbers should appear only in annotated genomes, and no such numbers should exist in MetaCyc. However, many such numbers are still found in MetaCyc due to the fact that in the past they were used where M-numbers should be used. They are being slowly converted to other types of EC numbers.

Curation Guidelines

- It is very important that a curator is familiar with the EC system. When entering a new reaction, the curator needs to evaluate whether it fits an existing EC number (as either official or unofficial reaction). If not, the curator should assign a new proprietary M-number to the reaction (see below).

- The EC system is represented in the database as a set of EC-Number objects (that are different from reactions). Since the information associated with an EC number (enzyme name, references, comment etc.) is already present in the EC-Number page, you should not duplicate this information in the reaction page.

- Reactions are connected to EC numbers by typing the number in the reaction editor. You can assign as many EC numbers as appropriate. For each number you must indicate whether it is an official EC number. The EC number is official only if the reaction is in the exact form as defined by the Enzyme Commission.

- You may want to give a reaction an “unofficial EC number” if the reaction is clearly catalyzed by the same active site of the same enzyme described by the EC entry, and fits the definition of that entry, yet differs in the reaction equation. For example, EC 2.7.1.1 defines the enzyme hexokinase, and the official EC reaction uses the class “D-hexose” as the substrate. An instantiated reaction that utilizes the specific sugar α-D-glucopyranose clearly applies to this enzyme, yet differs in the reaction equation, and thus should be given 2.7.1.1 as an unofficial number.

- You should assign an unofficial EC number to a reaction only if you believe that PathoLogic should assign any enzyme that is annotated by this number to this reaction. Otherwise, generate a new M-number for the reaction.
- If a reaction is a sub-reaction of a composite reaction with an EC number, and does not have an EC number on its own, it should not be given any EC number (since the EC number describes the over-all reaction, not the partial reaction). When the reactions that make up the composite reaction are shown in a pathway diagram, they are annotated by the EC number of the composite reaction, surrounded by square brackets.

M-numbers

- If the reaction does not fit any formal EC number, the curator should assign a proprietary M-number to the reaction (e.g. 2.1.1.M7) by clicking the "Assign M number" button in the reaction editor.
- Note that assigning a new M-number results in the creation of a new EC-number object for that number (e.g. EC-2.1.1.M7). The curator should assign a common name to the new entry, and ideally provide a comment, alternative names, and citations in the same format as regular EC entries. Editing the EC number object is accessed by right-clicking the EC number and selecting Edit → EC-Number Editor.

Names

- A reaction name should be entered only for reactions that do not have an EC number. Typically reaction names are of the same form as enzyme names, and are particularly important to provide if there is no associated enzyme. For transport reactions Pathway Tools automatically generates a name such as "Transport of pyruvate". Therefore a common name should only be provided for transport reactions if the generated name is incorrect or insufficient.

Comment

- A comment is not necessary for reactions with an EC number, since the relevant information is available at the EC number page. However, for reactions that do not have an EC number, it could be useful to add a comment, especially if there is something unusual about it. If the reaction is hypothetical, explain the supporting evidence.

Citations

- Citations are not necessary for reactions with an EC number (since the EC entry includes the relevant citations), but are **mandatory** for new reactions without an EC number. Always provide a citation documenting the source of information supporting the existence of the reaction.

Reaction Directionality

- The direction in which the Enzyme Commission writes reactions does not imply the direction of the enzymatic activity (note that the EC uses the equal sign rather than an arrow in the equation), and was devised only to standardize the description of all the enzymes in a particular sub-subclass. This direction often differs from the direction in which a specific enzyme operates physiologically. Thus, it is very important to indicate the direction of the reaction.

- If the reaction is known to be reversible or unidirectional inside the cell, this information should be specified using the reaction editor. For reversible reactions, it is quite possible that enzymes from different organisms (or even different compartments within the same organism) will catalyze the same reaction in different directions. In such cases the reaction should be marked as reversible in the reaction editor, while the directions for specific enzymes can be specified in the protein editor in the enzymatic activity section.

Spontaneous reactions

- If a reaction occurs without enzyme catalysis, this should be indicated by selecting this check mark. This information will be shown on reaction and pathway pages.

Physiologically Relevant Reactions

- Some reactions involve substrates that are unlikely to be encountered by an organism. If this is a case, make sure to deselect the button “Physiologically-Relevant?”, which is selected by default. One reason to specify a reaction as not physiologically relevant is so its substrates will not be identified as dead-end metabolites by the dead-end metabolite finder.

Reaction Locations

- When a new reaction is created, its location is specified by default as the cytoplasm. If a reaction is known to be located at a different cellular location (e.g. only in the Golgi apparatus), the location information should be modified to the correct value. When the reaction is known to be present in multiple cellular locations, this information should be

entered by clicking the button “Add Additional Reaction Location” in the reaction editor. This also applies to transport reactions – multiple sets of locations can be defined within the reaction editor.

Enzymes Not Used

- The Pathway Tools software often makes inferences about which enzymes catalyze which reactions, based on the use of compound classes in generic reactions. Sometimes the curator wishes to specifically prevent the attachment of a particular enzyme to a particular reaction. To do that, open the reaction frame in the Frame Editor, and add the frame IDs of the enzymes to be excluded to the slot Enzymes-Not-Used.

3.3. PATHWAYS

This section discusses the curation of pathways.

3.3.1. SUMMARY OF INFORMATION COLLECTED FOR PATHWAYS

- Common name
  - Special types of pathways are those classified under the activation/ inactivation/Interconversion class. In these cases, the pathway should be named after the better known compound. For example, when an activation pathway describes the conversion of A (inactive) to B (active), the pathway should be named B activation, because B is the well-known compound. However, A activation should be entered as a synonym.
- Synonym(s) of pathway name
- Superclass(es) of pathway
- Evidence code for the existence of the pathway, along with a reference
- Names of species in which the pathway has been experimentally demonstrated (only relevant to MetaCyc since other PGDBs are specific to a given organism).
- Summary
  - General description of the pathway and its significance.

  - Please try to use uniform styling for the General Background and About This Pathway headings within summaries. Consistently use these two phrases, in bold, with a blank line before the next sentence. No underlining, no colon.
  - Statement regarding the initial and end substrates. In cases where the initial or end compounds are rather specific (as in the degradation of a xenobiotic compound or the biosynthesis of a secondary metabolite), provide background information about the compound. If the pathway is defined as the degradation of substrate A to substrate E, but E is further degraded, comment on how E is further degraded and to what end products in the species noted.
  - Statement regarding whether the pathway is shared among different types of species. – Relationship to similar pathways in the same or different species.
  - Relationship to linked pathways (preceding and subsequent pathways), and sub- and super-pathways (see definitions), if applicable. Use internal links when referring to other pathways.
  - Highlight interesting or novel reactions/enzymes in the pathway.
  - If the pathway contains proposed intermediates, or hypothetical reactions, discuss the relevant circumstantial or preliminary evidence.
- Links to other pathways within the same PGDB.
- Links to other DBs, such as other PGDBs or The University of Minnesota Biocatalysis/Biodegradation DB (UM-BBD).
- Label hypothetical reactions as such.
- Citations

Pathway Naming. When adding a new pathway name, try to use the format and style of other, similar pathway names. For consistency, please use the terms “degradation” and “biosynthesis” for pathway common names when appropriate. Do not use the terms “catabolism”, anabolism, nor “utilization”. If “degradation” or “biosynthesis” is not appropriate, “metabolism” may be used.

Pathway variants should be enumerated with roman numerals, not cardinal numbers. For example, TCA cycle I, TCA II, etc.

Begin the name of any superpathway with “superpathway of...”

Spell out the full names of amino acids or chemical compounds within all pathway names.

3.3.2. PATHWAY DEFINITIONS

A metabolic pathway is a set of one or more enzymatic transformations (such as biosynthesis, degradation, conversion, or utilization), as it occurs in a particular organism. Identical pathways that exist in other organisms are not repeated in the MetaCyc database; instead a single pathway is labeled by the multiple organisms in which it occurs. Metabolism of a substrate exogenously supplied to cells, such as a vitamin, or drug, can also constitute a pathway.

Pathways can be classified into base pathways and superpathways. A base pathway is considered a lowest-level pathway in the sense that it is not subdivided into smaller component pathways. Base pathways can be linear, circular, or branched.

A superpathway is an aggregation of two or more pathways that are related in some way. A pathway component of a superpathway is referred to as a subpathway. A subpathway is part of a superpathway, and a superpathway is composed of subpathways. The subpathways of a superpathway can be base pathways, or can themselves be superpathways. Some superpathways will contain additional reactions and enzymes not found within the base pathways, such as reactions that connect two base pathways together. PGDBs always contain links between associated base pathways and superpathways, and those links are displayed by the Pathway/Genome Navigator toward the bottom of a pathway display page.

There are two main types of superpathways: those whose subpathways are related by a common substrate, and those whose subpathways are related by being analogous base pathways from different organisms. For the first type of superpathway, its subpathways could be derived all from the same organism, or they could be derived from multiple organisms. Multispecies superpathways that are connected via a common substrate are potentially useful in metabolic engineering. The steps used in creating superpathways from subpathways are described in the Pathway Tools User’s Guide, Volume II, section 2.3.5.3.

More specifically, superpathways can be created based on the following types of relationships among their subpathways: (1) subpathways that are physically connected through a common substrate (that is, one subpathway produces the substrate, and another subpathway consumes it); (2) subpathways that are unconnected, but that metabolize the same substrates (e.g. MetaCyc Superpathway of aspartate and asparagine biosynthesis: interconversion of aspartate and asparagine); and (3) analogous subpathways consisting of an analogous series of reactions

catalyzed by the same enzymes, or by analogous enzymes (e.g. MetaCyc Superpathway of isoleucine and valine biosynthesis).

Additionally, pathway variants exist in which the same substrate is synthesized or degraded using different enzymes, cosubstrates and/or cofactors, in the same or different organisms. Pathway variants share iden¬tical pathway names followed by a roman numeral. Many examples of pathway variants can be found in MetaCyc by browsing the pathway ontology. These pathway variants can potentially be combined into superpathways.

3.3.3. DEFINING PATHWAY START AND END POINTS

Several considerations guide the questions of how to define the start and end points of a pathway, and of whether a given published pathway should be encoded in a PGDB as a single base pathway, or as a set of base pathways within a common superpathway. The following rules should be used to guide the creation and editing of base pathways and superpathways, when possible.

- The substrate biosynthesized or degraded by a pathway should be a stable substrate, as opposed to a transient intermediate. However, a pathway could show the biosynthesis of a stable intermediate that is a precursor for the biosynthesis of other substrates.
- Biosynthetic pathways should begin with an intermediate of central metabolism. These intermediates are the 13 precursor metabolites: glucose 6-phosphate, fructose 6-phosphate, ribose 5-phosphate, erythrose-4-phosphate, triose phosphate, 3-phosphoglycerate, phosphoenolpyruvate, pyruvate, acetyl CoA, alpha-oxoglutarate, succinyl CoA, oxaloacetate, and sedoheptulose 7-phosphate. A pathway link should be created to indicate the pathway that produces the precursor metabolite at the start of the pathway.
- Degradative pathways that produce an intermediate of central metabolism should stop at that point. Some degradative pathways may not produce intermediates of central metabolism, but instead produce compounds that are excreted from the cell. If appropriate, a pathway link should be created to indicate the pathway that processes the resulting metabolite at the end of the pathway.
- Another class of pathways is applicable in cases where compounds are metabolized in a dissimilatory manner for the production of energy. Examples for these pathways include sulfate reduction and ammonia oxidation. In such cases, the metabolites are unlikely to consume or produce intermediates of central metabolism. Such pathways should start with the natural form of the compound being used as an electron donor or acceptor, and end

with the compound generated at the end of the electron transport process, which would generally be secreted by the organism.

- Very large or complex pathways should usually be defined as superpathways that combine several smaller base pathways, where those base pathways are divided at breakpoints. Dividing a large or complex pathway in this fashion is particularly useful to optimize the accuracy of PathoLogic predictions, especially in cases where it is the base pathways, rather than the entire pathways, that tend to be present as units across different organisms. If the pathway was defined as one large base pathway, rather than as a set of base pathways connected through a superpathway, PathoLogic would be unable to predict the presence of the smaller base pathways independently in different organisms. Breakpoints for large base pathways can be chosen based on various criteria such as: branch point substrates; substrates involved in regulation; a major metabolite that is further metabolized; the cellular compartment in which the reactions occur (organelle or cytosol); a transport segment or a utilization segment.
- If a published pathway contains several pathways that are already defined in a PGDB as base pathways, it should be represented as a superpathway.
- If the pathway contains too many reactions to conveniently represent in one base pathway, it should be broken into two or more base pathways, which should be linked together by pathway links (see below).

3.3.4. PATHWAY LINKS

Pathway links are a mechanism for indicating substrate connections among pathways. Pathway links are displayed as arrows connecting an input or output substrate in a pathway to the name of a second pathway in which that substrate is metabolized. Pathway links can illustrate the source pathway for an input substrate, or the destination pathway for an output substrate (assuming that it is not completely metabolized). Clicking on the second pathway name takes the user to that pathways display page. Links can be created to another base pathway, a superpathway, or to a class of substrates that derive from the pathway (e.g. MetaCyc glycolysis I).

Note that although it is helpful to explain the origin or fate of substrates in the pathway summaries field, this unstructured text is not computationally useful, and thus cannot replace the use of pathway links.

3.3.5. LIMITATIONS

Metabolic pathways involving macromolecules and cellular structures may be difficult to represent in PGDBs. This is a factor in pathway selection. Pathways that involve reactions that synthesize, degrade, or modify small molecule components of macromolecules and cellular structures can be represented. However, some processes may be beyond the scope of PGDBs, which focus on small molecule metabolism.

3.3.6. DATABASE SEARCHING STRATEGIES

To search available databases for information regarding a particular pathway, it is recommended to begin by using general keywords related to the pathway name (e.g. creatinine degradation to formate, methionine biosynthesis, etc.). You may need to search using several alternative names, for example: toluene degradation, toluene oxidation, toluene catabolism etc. Adding additional search terms such as anaerobic will help avoid getting irrelevant hits. As in all searches, if you get too many hits you should narrow your search by adding keywords, such as bacteria to limit the search to only bacterial pathways, or a species name to limit the search to only pathways in a particular species. Some databases allow the use of wildcards, which are truncated names followed by a special character such as * to designate different variations for the ending of the word. For example, bacter* would include bacteria, bacterial, bacterium, etc. Different databases may use different wild cards, so it is always useful to consult the databases search description/overview. For a description of PubMed searching strategies see the following URL:

http://www.ncbi.nlm.nih.gov:80/entrez/query/static/help/pmhelp.html#PubMedSearching.

If you know a little bit about the pathway, such as the names of intermediates (unique intermediates may be best to use), or enzymes involved in the pathway, you should also search for articles using these keywords. Furthermore, if you know the names of the researchers who studied the pathway, you can search for articles using a combination of their names and the substrate or enzyme they are working on. Once you get some articles of interest, you would usually find other related articles by 1) looking at their references, and 2) using SciSearch to find articles that cite them. Often an article's full text is available online in HTML format. These HTML formatted articles often have web links to articles that they reference as well as to articles that cite them, which is a very convenient way of finding additional articles. Once you have identified and gathered the relevant articles for a pathway, try to find their PubMed ID (PMID) numbers and label them as these numbers are the easiest way to enter references information into PGDBs.

3.3.7. PATHWAY ENTRY

Once you have enough papers to put a pathway together, draw out the pathway, making note of the chemical reactions, chemical names and structures of the metabolites, and enzyme names if known. Try to identify any EC numbers that may have been assigned to the reactions in the pathway. It is very important to find out whether the chemicals and/or reactions already exist in the database. This may not be straightforward, as chemicals may have many different names. MetaCyc already includes all of the reactions that have been assigned EC numbers, so you may want to search them carefully to find out whether existing EC reactions fit any of the reactions in the new pathway. Often authors are not aware of such EC numbers and do not include them in their publications. Make sure you do not create duplicate chemicals and/or reaction in the database, as this will lead to certain problems in the future. After you have identified all existing reactions, you may need to create new reactions and chemicals. Write down the frame IDs of both existing and new reactions and add them to the drawing of the pathway that you prepared. This will greatly facilitate the creation of the new pathway.

Once you finished these steps, you are ready to define the new pathway. Do not forget to assign the appropriate class and evidence codes. An ideal reference for the pathway evidence code is a recent review article that cites all the relevant experimental literature. In such case use the code EV-EXP-TAS. Make sure you assign the appropriate organisms to the pathway, and mark any hypothetical reaction as such. Once the pathway is defined in the PGDB, you need to enter enzymes and genes for the various reactions. Review the papers in greater detail while taking notes of the relevant information. As mentioned earlier, it is best if you are already aware of the type of information you'll input into the PGDBs. This way, you can skim/read the papers for the relevant information, and take notes in a similar fashion to how the PGDB is organized, which expedites inputting the information into the database. In addition, you'll need to cite the information you input into PGDBs. Hopefully, your papers have PMID reference numbers; in which case, the full reference information will be imported automatically

3.4. GENES

This section discusses the curation of genes.

3.4.1. SUMMARY OF INFORMATION COLLECTED FOR GENES

– Common name and synonym(s)

– Superclass(es) of gene

- Gene product type (eg. enzyme, regulator, leader, etc.). Evidence for product type (experimental or predicted based on sequence analysis)
- Transcription direction (unspecified, forward or reverse)
- Left and right end position of gene on chromosome or plasmid
- Link to other DBs
- Summary
- Citation(s)

3.4.2. GENE NAMING

Regarding gene naming, for bacterial databases, automated programs will periodically ensure that the capitalized gene name is a synonym for the name of the gene product (e.g., “TrpA” will become a synonym for the product of “trpA”). For most organisms, the unique identifier assigned to the gene by the genome project (e.g., “HP0001” for an *H. pylori* gene) will already be present in the PGDB as the frame name or in the slot Accession-1 or Accession-2. Therefore, there is no need to add this same identifier as a synonym for the gene name.

3.5. PROTEINS

This section discusses the curation of proteins.

3.5.1. SUMMARY OF INFORMATION COLLECTED FOR PROTEINS

- **General Protein Information**
  - Species name (in multi-organism PGDBs)
  - Common name and synonym(s) of the protein
  - Cellular location (e.g. membrane, cytoplasm, chloroplast, etc.)
  - Observed and calculated molecular weight in kilodaltons
  - Summary (see section 3.5.2 below)
  - Citation(s)

- Last-Curated. The last-curated date is the date on which a systematic literature search was last performed by curators for this gene product. This date can be used by both curators and database users to determine how up to date the entry is. Checking the "last-curated" box in the protein editor will cause this field to be updated. Do not check this box if only partial curation is performed by the gene product, as this would interfere with the purpose of this field, which is the record the last date on which full curation was performed.
- Neidhardt Spot Number (reflects the proteins electrophoretic behavior in 2-dimensional electrophoresis). (Can be added via the frame editor, but not typical. Add it if you come across this information, but do not search for it).

- **Enzyme Activity**
  - Enzyme activity name and synonym(s). This name should be based solely on the catalyzed reaction. Additional information, such as a roman numeral used to identify a particular isozyme, should be entered under the enzyme common name, not here (see Enzyme Naming below).
  - Summary. General description of the enzymatic activity, highlighting any interesting aspects. Kinetic data that does not have a dedicated slot should be noted, if available.
  - Inhibitors (physiologically relevant or not?)
    - Competitive inhibitors: inhibit enzyme activity by binding reversibly to the enzyme and thereby preventing the substrate from binding.
    - Noncompetitive inhibitors: inhibit enzyme activity by binding reversibly to either the free enzyme or the enzyme-substrate complex. The substrate is not prevented from binding, but the enzyme with the inhibitor bound is not catalytically active.
    - Uncompetitive inhibitors: inhibit enzyme activity by binding reversibly to the enzyme-substrate complex.
    - Allosteric inhibitors: inhibit enzyme activity by binding reversibly to the enzyme and inducing a conformational change that decreases the affinity of the enzyme to its substrates. Allosteric inhibitors can be competitive or noncompetitive, therefore those inhibition categories may be used in conjunction with this one.
    - Irreversible inhibitors: irreversibly inhibit the enzyme activity by binding to the

enzyme and dissociating so slowly as to be considered irreversible.

- Other inhibitors: inhibit the enzyme activity by a mechanism that has been characterized but does not fall cleanly into one of the above categories.
- Inhibitors of unknown mechanism: inhibit enzyme activity, but the mechanism of action is unknown either because it has not yet been elucidated, or because it has not been curated.

- Activators (physiologically relevant or not?)
- Cofactor or prosthetic group. Cofactors and prosthetic groups are non-protein chemical compounds that are required for the enzyme's activity. They can be as simple as a magnesium ion and as complex as cobalamin. Prosthetic groups are defined as being covalently or tightly bound to the enzyme, whereas cofactors are not. Since it is not always possible to define what is tightly bound and what is not, both types are curated under the same category in MetaCyc. Unlike substrates and cosubstrates, cofactors remain in the same form after completion of the reaction, ready to catalyze another round.
- Alternative substrates: Can be used to specify other known substrates for the enzyme. In general it is preferable to define reaction frames for alternative substrates. Curate alternative substrates using the “Alternative substrates” field within the enzyme activity only when the full reaction equation is not known or cannot be easily inferred.
- Citation(s).

- Enzyme Subunit Composition
  - Subunit composition. Specifies the number of copies of each monomer subunit of a multimeric protein. In cases where sub-complexes of a large multimer have been observed, those sub-complexes can be created as PGDB objects that are sub-complexes of a larger super-complex.
  - Subunit name(s) and synonym(s)
  - Subunit molecular weight (experimental and computed from sequence)
  - PI (isoelectric point). Can be added via the frame editor. Add it if you come across the information but do not search for it.

    - UNIPROT primary accession number of subunit, if available (http://uniprot.org/). Links protein subunit to UNIPROT entries.
    - Summary
- An optional brief description of the particular subunit, such as its function, if known or proposed. For example, a subunit may be known to house the catalytic active site, it may have an FAD binding motif and may be proposed to be involved in electron transport, or it may be proposed to be the membrane anchor subunit of a membrane-bound enzyme.
    - Citation(s)

3.5.2. SUMMARIES

Summaries should be stored in the PGDB objects for the protein or RNA product of a gene, rather than in the gene itself.

The first sentence of the summary should summarize the function of the gene product.

Protein summaries should describe important information known about the protein such as its function and role within the cell, any phenotypes resulting from its mutation or absence, protein domains, its participation as a component of a larger structure, its similarity to other proteins (including any functional complementation studies), membership in protein families, and information about its structure. If the enzyme is novel, explain why. If the enzyme has different isoforms, describe the substrate specificity of the isoforms and the cell-type/tissue/developmental specificity of the isoforms.

Except in rare cases, in which the gene product is particularly complex or physiologically significant, an effort should be made to keep the length of “Summaries” (including references) to less than 500 words. In longer summaries, the first paragraph should in a few sentences summarize what is known about the gene product. Experimental support for these conclusions and information regarding protein structure and regulation of expression should come later. In other words, “Summaries” should be organized more like a news story than a scientific paper: conclusions should precede rather than follow detailed information and evidence; the user should be able to gain the essential facts without necessarily having to read the entire summary.

In those cases in which background information is considered to be an important aid to some users, such information should be placed in a separate section under the heading “General Background”. In such cases the information about the specific protein should be entered under

the heading “About This Protein”.

It is helpful to explain gene names within summaries. Use consistent styling including double quotes and bold characters to indicate what longer phrase a gene name abbreviates. Where the explanation is put will vary between proteins. Sometimes it will work best at the start of the summary, for example:

“N-**a**cetyl**g**alactos**a**mine repressor,” AgaR, controls the expression of the aga gene cluster...

In cases where the gene name is not an abbreviation of the protein name, the explanation will often work best at the end of the summary, in a paragraph of its own. For example:

Apt: “**a**denine **p**hosphoribosyl**t**ransferase” [Kocharian76]

Literature citations are an important and valuable part of summaries, but some effort should be made to restrict their numbers to the most significant ones — try not to exceed 10–20 references within most summaries (in some cases of extremely well-studied genes, more references may be appropriate). Sentences, particularly those in the first paragraph, should not be interrupted by long listings of tens of references.

If a literature search fails to turn up any information about a particular object, a statement with the following exact wording should be included in the summary (of course, change the date as appropriate): “No information about this protein was found by a literature search conducted on 18 June 2003.”

3.5.3. ENZYME NAMING

Very often the names used by the scientists who describe an enzyme are not appropriate. If an EC number has been assigned to the enzyme, use the “accepted name” in the EC entry for the enzyme’s common name (and/or enzymatic activity). If not, try to see how similar enzymes have been named by the EC. UniProt often has better names than those reported in the original literature. Keep in mind that unlike UniProt, BioCyc tier 1 PGDBs do not capitalize the first letter in the enzyme name.

Non-specific names: names such as “oxidoreductase” or “hydrogenase” are non-specific, because they refer to nonspecific classes of enzyme activity, not to specific enzyme activities. Use of these names in MetaCyc could be very problematic for the PathoLogic program, because genome annotations often use these nonspecific enzyme names for genes whose specific functions cannot be inferred. Thus, if a MetaCyc enzyme, assigned to a reaction, was called “oxidoreductase,” PathoLogic would assign all genes annotated with that name to the

corresponding reaction, which is not correct.

Never assign non-specific enzyme names as the common name. If you must use one because it is an accepted name in the field, add it as a synonym for the enzyme name. Synonyms for the enzyme names will be screened for non-specific names by the PathoLogic Name Matcher, and removed if not specific.

The word “monomer” should be used to refer to a polypeptide that acts as a member of a homo-multimeric complex. The words “subunit” or “component” may be used to refer to a member of a hetero-multimeric complex.

Avoid extra wordiness (“complex”, for example) unless it is necessary to avoid confusion. For example, the long name of the pyruvate dehydrogenase multienzyme complex distinguishes it from pyruvate dehydrogenase, which is one of its components.

The following procedures should be followed regarding entry of enzyme names ending in roman numerals and Arabic numbers, such as “pyruvate kinase II” and “tagatose 1,6-bisphosphate aldolase 2.”

3.5.4. ENZYME NAME VS. ENZYMATIC ACTIVITY NAME

PGDBs allow two sets of names and synonyms to be defined for enzymes: names for the enzyme activity(ies), and names for the enzyme itself. Consider that *E. coli* has two proteins with pyruvate kinase activity, designated pyruvate kinase I and pyruvate kinase II. The name of the enzyme activity for both of these enzymes is pyruvate kinase, however, the names of the enzymes are pyruvate kinase I and pyruvate kinase II.

To encode this situation in a PGDB, enter the enzyme name (e.g. pyruvate kinase I) in the top section of the protein tab of the protein editor, using the button “Edit Enzyme Name”. Enter the enzyme activity name, pyruvate kinase, under the Enzymatic Activity tab of the protein editor, in the box labeled “Enzyme activity name.” Enzyme activity names are mandatory, while enzyme names are optional.

Note that if the enzyme catalyzes multiple reactions, and different enzymatic activity names were entered, but no enzyme name was defined, the enzyme name would be displayed by concatenating the enzymatic activity names. However, if an enzyme name has been defined, it will be used instead of the concatenated name.

Note: Roman numerals in the enzyme names do not always describe different isozymes! For example, consider the enzymes exodeoxyribonuclease I and exodeoxyribonuclease III. These

enzymes catalyze different reactions, and the roman numerals designate exactly which enzyme activity an enzyme has. Therefore, these names should be entered in the "Enzyme activity name" field of a PGDB, because the names are specific to the enzyme activity, not to the protein.

Assigning the correct enzymatic activity name: It is important that the enzymatic activity name is accurate and descriptive. If the enzyme has been classified by the EC, and the reaction catalyzed is the official EC reaction, it is often useful to use the EC "Accepted name" for the enzymatic activity name. If the reaction has an unofficial EC number, the EC accepted name should be revised accordingly. For example, the accepted name for EC 1.3.1.22 is "3-oxo-5α-steroid 4-dehydrogenase (NADP+)", and the official EC reaction uses the generic substrate "a 3-oxo-5α-steroid". For the instantiated reaction that uses progesterone, which has an unofficial EC 1.3.1.22 number, the proper enzymatic activity name is "progesterone 4-dehydrogenase (NADP+)".

If the enzyme has not been classified by the EC, it is often useful to look at related enzymes (that belong to the same sub-subgroup) and try to assign an enzymatic activity name that is based on the accepted names of those related enzymes.

3.5.5. MODIFIED PROTEINS

Modified forms of proteins can describe assorted post-translational modifications to the protein (e.g. phosphorylated forms). The protein editor allows the curator to create as many modified forms of the protein as required. When creating a modified form, the system automatically creates the common name "Modified xxx" (when xxx is the common name of the unmodified form). The curator should change this name to be descriptive of the type of modification, and provide a short summary explaining relevant information. When the editor is closed, the modified form(s) are listed underneath the summary of the unmodified form and vice versa. By right-clicking these modified forms and selecting Show -> Show frame name, the frame ID of this modified form is printed in the lisp window, enabling the curator to use it in reactions if required.

A special case of modified enzymes describes proteins that are cleaved post translationally, followed by interaction of the two fragments to form a complex, which is the active enzyme (e.g. adenosylmethionine decarboxylase, speD). To describe this process, the curator should generate modified forms for each fragment. A new complex is then created from the modified forms, and the enzymatic activity is assigned to the complex.

3.5.6. LOCATIONS OF TRANSPORTED SUBSTANCES

Transporters from different organisms, or from different membranes within the same organism, can often transport the same compound across different membranes. When a transport reaction is known to occur across multiple compartments, these locations should be specified for the reaction using the reaction editor. When a protein has been assigned to a transport reaction with multiple locations, a selector appears within the protein editor at the enzymatic activity section, letting the curator choose the appropriate location for the specific enzyme.

3.6. RNAS

This section discusses the curation of RNA gene products.

3.6.1. TRNA NAMING

Organisms generally contain multiple copies of tRNAs that have the same amino acid specificity, but recognize different anticodons. The naming convention for tRNA products follows the format “tRNA-AminoAcid(anticodon)”, for example, tRNA-Ala(UGC). The anticodon sequence is often the source of confusion and errors. For this reason, as of 2017, NCBI does not accept the addition of the anticodon sequence to the tRNA name in GenBank files. However, properly verified anticodon sequences are useful to users and can be included in the name. The anticodon is the sequence actually present in the tRNA itself; that is, if the codon sequence in an mRNA is 5’-GCA-3’, the anticodon sequence present in the tRNA is 5’-UGC-3’.

4. COORDINATING CURATION BETWEEN METACYC AND OTHER TIER 1 PGDBS

4.1. INTRODUCTION

A critical curation concept is the concept of **the authoritative PGDB** for a given database object. The same object (e.g., a reaction, metabolite, or pathway) can be found in multiple PGDBs because PathoLogic copies reactions, metabolites, and pathways from MetaCyc to a new PGDB when a new PGDB is created. The authoritative PGDB for an object is considered to contain the master version of the object, from which other versions are updated. It is important that curators know which PGDB contains that master version, and how to move objects between PGDBs to maximize curator efficiency. The rules in this section apply to curators who work in the SRI Bioinformatics Research Group, and collaborating curators who have update access to MetaCyc and to other PGDBs.

- **For compounds, reactions, EC numbers, and pathways, MetaCyc is the authoritative PGDB.**
- **For enzymes and genes, the Tier 1 PGDB for that organism is the authoritative PGDB. If no Tier 1 PGDB exists for that organism, then MetaCyc is the authoritative PGDB.**

Example: EcoCyc is the authoritative PGDB for the enzymes and genes of *E. coli* K-12 MG1655. However, MetaCyc is the authoritative PGDB for the pathways, reactions and metabolite objects within EcoCyc.

Several times a year, prior to a new release of Pathway Tools or BioCyc, a propagation process synchronizes common entries among tier 1 PGDBs. For each object that is propagated the data in one or more PGDBs is replaced by the data from the authoritative PGDB. If updates are not performed in the authoritative PGDB then this process will sometimes clobber updates made in the non-authoritative PGDB, and thus it is important that all curators familiarize themselves with this process and are aware of the proper procedures.

When editing existing objects, it is very important that the changes are made in the authoritative source PGDB, ensuring that these changes will be propagated in the correct direction when propagation occurs. In addition, when modifying enzymes that appear in MetaCyc but originated from other tier 1 PGDBs, it is always a good idea to invoke limited propagation of the object to MetaCyc following the change, making sure that the object is identical between the organism-specific PGDB and MetaCyc, thus eliminating any possibility of data loss during the next propagation event.

In the sections that follow we explain these rules in more detail and elaborate on the protocols for transferring objects among PGDBs.

4.2. CREATING AND UPDATING PATHWAYS

MetaCyc is considered the authoritative PGDB for all pathways that are included in it (including legacy pathways that were originally created in different PGDBs).

- All curators should create new pathways and revise existing pathway diagrams in MetaCyc since the collection of chemical compounds, reactions, and EC number entries in MetaCyc is typically more up-to-date than in other PGDBs.

4.2.1. CREATING A NEW PATHWAY

- If the curator wishes to assign enzymes from an organism for which a tier 1 PGDB exists (as of 2013, the list consists of *Escherichia coli* K12*, Homo sapiens, Arabidopsis thaliana, Saccharomyces cerevisiae, and Bacillus subtilis*), it is important that the enzymes are shared between the organism-specific PGDB and MetaCyc (meaning they have the same frame ID in both databases, and their data is identical).

The recommended procedure is as follows:

1. Export the relevant enzymes from the organism-specific PGDB to MetaCyc by right-clicking the enzyme and selecting Edit → Export to DB, then selecting MetaCyc.
2. Create or open the pathway in MetaCyc.
3. In MetaCyc, assign the enzymes to the appropriate reactions, and perform the curation.
4. When all curation is completed, export or propagate the pathway to the organism-specific PGDB (by right-clicking on the pathway name and selecting Edit → "Propagate pathway and enzymes to DB", then selecting the other PGDB).
5. If the MetaCyc pathway contains additional enzymes from other organisms, propagate the pathway by using the command "Propagate pathway to DB", followed by propagation of the individual enzymes from MetaCyc to the original PGDB by right-clicking the enzyme and selecting Edit → Propagate enzyme to DB, then selecting the other PGDB.

- A cautionary note: the enzymes in the organism-specific PGDB may be associated with additional reactions, and if the enzyme has not been curated previously, these associations may be incorrect. Once you propagate the enzymes back to the original PGDB, make sure that no incorrect associations exist there, and if they do, delete them (by right clicking on the enzymatic reaction name and selecting Edit → delete Frame).

- Pathway comments are not propagated among PGDBs. By default, if the pathway in the organism-specific PGDB does not contain a comment, the MetaCyc comment shows up. If curators wish to write different comments for an organism-specific PGDB, the comment should be written in that PGDB. It will not be propagated into MetaCyc, nor would it be overwritten by the MetaCyc comment.

- Curators of organism-specific PGDBs are reminded that MetaCyc pathways contain some fields not found in organism-specific PGDBs. Two of those are a taxonomic range for the pathway and a list of species. In MetaCyc enzymes are shown in pathway diagrams only for organisms listed. Make sure that you add this information to the newly created pathway. These fields would not be propagated to the organism-specific PGDB. In addition, if the pathway is a variant of a similar pathway already in the database, and differs by one or more reactions, you should specify the differing reactions as key reactions in both pathway variants. Doing this greatly helps the PathoLogic component of Pathway Tools to select the correct variant when creating a new PGDB.

4.2.2. ALTERING A PATHWAY IN AN ORGANISM-SPECIFIC PGDB

- If a pathway in an organism-specific PGDB should be altered from the MetaCyc version, the procedure should start by right-clicking on the pathway name and selecting Edit -> Duplicate Frame and Edit. This will create a new pathway, identical to the original one, but with a new frame ID. At this point the original pathway can be deleted, and all modifications should be performed on the newly created pathway. Having a different frame ID ensures that the changes would not be overwritten by the MetaCyc pathway during a future propagation. If the curator decides that the new pathway should be present in MetaCyc as a new variant, the new pathway should be propagated to MetaCyc by selecting Edit -> Propagate pathway to DB, then select MetaCyc.

4.3. MODIFYING ENZYMES

The authoritative PGDB for enzymes from organisms for which a tier 1 PGDB exists is always the organism-specific PGDB. Updates to the enzyme should always be made in that PGDB, and propagated to MetaCyc if the enzyme exists in MetaCyc.

1. MetaCyc curators that need to update enzymes that were imported from another tier 1 PGDB should perform the update in the original PGDB, then propagate the modified enzyme to MetaCyc.

2. If the updating needs to be performed in MetaCyc (e.g. it requires attaching new reactions or references to compounds that do not exist in the organism-specific PGDB), start by propagating the enzyme from the organism-specific PGDB to MetaCyc, thus ensuring that the frames in the two databases are synchronized. Now make the changes in MetaCyc, and when done, propagate it back to the original PGDB.

3. In the case of EcoCyc, there is no need to propagate changes in protein curation immediately to MetaCyc, since data is propagated routinely from EcoCyc to MetaCyc several times a year.

5. METACYC-SPECIFIC INFORMATION

MetaCyc describes the union of pathways across a range of different organisms. This section contains information that is specific for MetaCyc curation, but may be applicable to other multi-organism PGDBs.

5.1. RESEARCHING PATHWAYS FOR CURATION IN METACYC

While researching a pathway you may find that the pathway was studied in many different species, just a couple, or maybe only one. It is a greater priority to add as many different pathways to MetaCyc than to find all of the species associated with a given pathway.

If the pathway has been studied in more than one species, find out whether a model species exists in which the pathway has been studied the most. If so, you may want to focus on this organism. However, quite often, different enzymes of the pathway have been studied in different organisms. When deciding which enzymes (meaning from which organisms) to curate, keep in mind that for an enzyme to be displayed in the pathway diagram, it has to be from an organism that is listed as a taxon in which the pathway occurs, and that a pathway can be associated only with organisms that are believed to contain all of the pathway's enzymes. You may create enzymes that catalyze a reaction even if those enzymes are from other species besides those specified for the pathway, but those enzymes will not be displayed within the pathway diagram.

If you find that different species use slightly different pathways, you should curate multiple pathway variants.

5.2. SPECIES INFORMATION

MetaCyc describes pathways for many different organisms. When entering a pathway into MetaCyc, you should record some species in which the pathway is known to occur in the Species slot of the pathway, which is done using the Species field of the Pathway Info Editor. The list of species recorded for a pathway should be considered a representative list, not an exhaustive list of all species in which the pathway occurs. However, if during your research you discover that the pathway is associated with a limited number of species, list all of them.

Naturally, if a pathway is known to occur in broad taxonomic groups, there is no point listing all of those organisms.

When creating the enzymes that catalyze steps in the pathway, specify the species from which the enzymes were obtained. Each enzyme in MetaCyc should define a protein from a single species.

5.2.1. TAXONOMIC RANGE INFORMATION

Should the curator be reasonably certain that a pathway should be expected to occur in a limited set of taxa (e.g., plants only, or animals only) a high-level taxonomic classification for its expected taxonomic range should be entered in the Taxonomic-Range slot of the pathway frame by using the Expected Taxonomic Range field in the Pathway Info Editor. Here you can limit the taxonomic range of organisms in which the pathway occurs by selecting the appropriate taxonomic domain(s) or subdomain(s). You can determine the most specific higher level domain for each genus and species by using the NCBI Taxonomy Browser. The higher level domain will usually be a phylum, or class. If the species is not present in the NCBI taxonomy, select only the highest level, a superkingdom (Archaea, Bacteria, or Eukaryota). Only enter this information if you are reasonably confident that the distribution of the pathway will be limited.

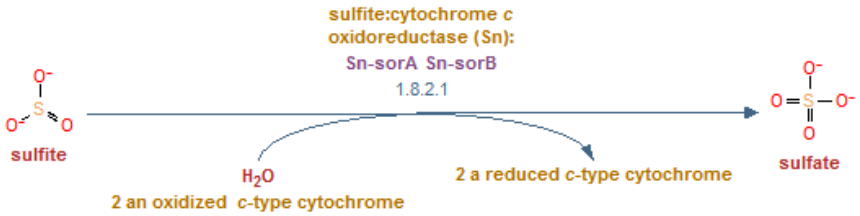
The Taxonomic-Range slot is primarily intended for use by the PathoLogic pathway prediction program during generation of new PGDBs. By assigning an expected taxonomic range to the organisms in a MetaCyc pathway, the domains, or subdomains that are represented in the expected taxonomic range can be compared with that assigned to the organism in the new PGDB. PathoLogic could then assign a lower probability score to the pathway if the organism for the PGDB is not in the expected taxonomic range of the MetaCyc pathway. This form of reasoning can help to exclude inappropriate pathways from the new PGDB. Expected Taxonomic Range also gives MetaCyc users a taxonomic perspective of the pathway that they might not otherwise have if they are not familiar with some of the organisms in the species slot.

5.3. PROTEINS AS SUBSTRATES IN METACYC

Some reactions in MetaCyc pathways involve protein substrates such as acyl carrier protein (ACP) or thioredoxin. These protein substrates should be entered as protein class frames within MetaCyc. The protein class frames are generic, species-independent descriptions of these proteins. The reason for this approach is that when MetaCyc pathways are predicted in organism-specific PGDBs, the reactions are copied over to that PGDB and are used in the

context of this specific organism, and must not refer to protein instances from other organisms (e.g. a human pathway utilizing an *E. coli* protein as a substrate).

If organism-specific instances of such proteins exist in MetaCyc (e.g. specific E. coli thioredoxins), these instances should be classified within the appropriate protein class. This will enable the user to see a list of all such instances by clicking on a protein substrate in a pathway diagram.



A reaction using protein classes

5.4. REFERRING TO ENZYMES IN PATHWAY COMMENTS

When referring to enzymes in MetaCyc pathway comments using internal hyperlinks, it is best to refer to the EC-number entry rather than a specific enzyme from a particular organism. Since EC entries are universal rather than organism-specific, this makes the comment much more useful when read from an organism-specific PGDB.

6. ECOCYC-SPECIFIC INFORMATION

6.1. E. COLI GENE FRAME NAMES

All frame IDs for *E. coli* genes are derived using numeric sequences. Some genes frame names are the same as the identifier used in Rudd's EcoGene database; they use the prefix "EG". Genes with prefix "G" were either (a) created by the MBL group when they encountered new genes in the literature, or (b) created by SRI — virtually all of these genes were created in the course of including data from the Blattner GenBank entry for the full E. coli sequence into EcoCyc. Examples: EG10115, G1001.

Identifiers used by other databases are often listed in the synonyms slot for the gene, e.g., the Blattner "b" identifiers and later assigned "EG" identifiers.

Dr. K. Rudd developed a naming scheme for E. coli ORFs, in which ORF genes are named beginning with "y." The rest of the name encodes the position of the gene on the E. coli chromosome. These names are retained as synonyms for genes whose functions are later determined.

6.2. INTERRUPTED GENES

Interrupted genes in E. coli are defined as follow. For each piece of the coding region, a separate gene frame is created. The same EG number is stored as a synonym for each interrupted gene, but the two interrupted genes have different b-numbers. In addition, a name of the form "ilvG" is stored as a synonym, but a name of the form "ilvG 1" is stored as the common-name. The interrupted? slot is set to T, and the start-base and end-base delimit the segment of the coding region defined by that gene frame.

6.3. THE MULTIFUN CLASSIFICATION SYSTEM

The genes in EcoCyc are classified using the MultiFun system developed by Monica Riley and her colleagues [5].

The gene class hierarchy has several classes specific for classification of uncharacterized genes. The general gene class "ORF" has three child classes as of the most recent updates to the hierarchy (in July 2003):

1. conserved ORF 2. conserved hypothetical ORF 3. nonconserved ORF

As we describe them to the users: “Conserved ORFs have homologs, usually in other organisms, where one or more of those homologs have known functions, but the sequence similarity of a conserved ORF to its homologs is not strong enough to permit inference of function of the conserved ORF. In contrast conserved hypothetical ORFs have homologs, but none of those homologs have known functions.” The remaining category, nonconserved ORF, is used to refer to genes that do not have assigned gene class functions and which do not have significant similarity to other genes.

As guidelines for naming of the products of these uncharacterized genes, we suggest that a product of a “conserved ORF” will be called “conserved protein,” a product of a “conserved hypothetical ORF” will be called “conserved hypothetical protein,” and a product of an “nonconserved ORF” will be called “hypothetical protein.” In cases where some additional information has been recorded, but this information is insufficient for prediction of protein function, additional information may be appended to the standard name; for example, “conserved protein with an unusual xyz domain.”

The explanations of selected gene class definitions are provided by Dr. Gretta Serres:

- The “cell structure” category contains both genes encoding components of the structure and genes encoding products involved in the biosynthesis of the structure.
- The terms “trigger” (3.4) and “modulator” (3.5) refer to specific compounds/proteins that either trigger a response or modulate a response. I, for example, in the case of lacI, lactose acts as the trigger and CRP acts as a modulator of the regulation.
- Any transcriptional activator or repressor should be assigned a class of 3.1.2.2 or 3.1.2.3, or both for dual regulators.
- The “adaptation to stress” class is intended to cover the classical inducers of stress such as osmotic pressure, temperature, and starvation.
- The “protection” class is intended to cover those inducers related to cell killing or stress induced by compounds or chemicals.
- The “defense/survival” category is intended to cover virulence factors.
- The “ extrachromosomal origin” category is intended to include chromosomal genes affecting functions of genes of extrachromosomal origin as well as genes that are of extrachromosomal origin.

7. UPDATE PROPAGATION AMONG DBS

Many DB updates (e.g., corrections to chemical structures) are non-species-specific, and therefore are intended to apply to all DBs. It is convenient to make these changes only once in one DB, and rely on an automated mechanism for propagating such updates to all the other DBs. In general, it is recommended that all such updates be performed in MetaCyc. Since this is not always convenient, the update propagation mechanism will take care of propagating certain kinds of updates from clone DBs (including EcoCyc) "up" to MetaCyc. However, all schema changes must be performed in MetaCyc otherwise they will be undone during update propagation. For instance-level updates, the update-propagation software uses the transaction logs collected in MySQL to determine when one updated value is more recent than another, to determine in which direction the update should be performed.

Update propagation is invoked manually and, for the time being, relatively infrequently. It is a time-intensive operation, which can take on the order of a half hour per DB. The update propagation mechanism does not save any of the updated DBs - it is up to the user to examine the output to make sure that it looks reasonable, and to save the affected DBs.

7.1. INVOKING DB UPDATING

To invoke the update propagation mechanism, call (propagate-kb-updates). This is the normal mode of update propagation and will cause updates to be propagated among all known DBs. Alternatively, to propagate only among a subset of DBs, the above function can be called with a list of orgkb-descriptors (an orgkb-descriptor can be found by calling find-org on the org-id). For example, to propagate updates only between EcoCyc and MetaCyc (MetaCyc is automatically always included in update propagation), you would type (propagate-kb-updates (list (find-org 'ecoli))).

Update propagation should be done with care, and the output examined fairly carefully before any DBs are saved (update propagation does not automatically save the changed DBs). There may still be bugs in the mechanism, and we want to take pains to avoid losing any data or undoing important changes.

7.2. OVERVIEW OF THE UPDATING PROCEDURE

Update propagation involves the following sequence of events:

- Any schema classes (classes can be explicitly specified to be either schema or data classes) in MetaCyc that are not present in the clone DBs are copied to the clones.
- All of MetaCyc's slotunits are copied over to all other clones. This will overwrite the correspond¬ing slotunits in the clone DBs. Slotunits present in a clone DB but not in MetaCyc will not be deleted, but a warning will be printed.

- All instances of the Databases class will be copied from MetaCyc to all clone DBs, overwriting the existing frames if they exist.
- The class hierarchy of each clone DB is "molded" to match that of MetaCyc, i.e. each class will have the same set of parents in all DBs. This operation doesn't copy any classes, it just potentially alters the parents of existing classes.
- For each clone DB, we loop through all instances of designated classes, looking at certain slots to see which updates need to be propagated "upwards" to MetaCyc. The set of considered classes and slots is listed below. The mechanism of this is as follows:
  - If a frame in a clone DB is not present in MetaCyc, and either it has never been deleted from MetaCyc or the creation date in the clone DB is more recent than the deletion date in MetaCyc, then the frame is copied to MetaCyc.
  - If the frame's parents are different in the clone DB and MetaCyc, and the last change to the parents is more recent in the clone DB than in MetaCyc, then the parents in MetaCyc are changed to match those in the clone DB. Any parents not present in MetaCyc that are necessary for this operation (presumably, these are all data classes) will first be copied to MetaCyc.
  - If, for each considered slot, the slot values and/or annotations are different between the two DBs and the modification date is more recent for the clone than for MetaCyc, then all slot values and annotations (except for citation and comment annotations, which are often species-specific) for that slot in that frame are copied from the clone DB to MetaCyc.
- While looping through these frames, we collect a list of frames and slots that either were updated in MetaCyc or that differ between the two DBs (and were not updated because the MetaCyc changes were more recent). This list is used in the next phase of the update.

- After updates have been propagated “upwards” from all clones to MetaCyc, we iterate through all the clone DBs again, propagating changes “downwards” from MetaCyc to the clones:
  - We “mold” all instances of the designated classes in the clone DB so that each instance’s parents match those in MetaCyc (because any more recent changes in parentage in any of the clone DBs have already been copied to MetaCyc, we cannot lose any changes here). If an instance exists in a clone DB, but its new parents do not (these would have to be data classes, otherwise they would already have been copied), we copy those parents over from MetaCyc.
  - We iterate through the list that we have collected of frames and slots that either have been updated in MetaCyc or potentially need updating in one or more clones, comparing values between the clone and MetaCyc. If there is any difference, the values and annotations (except for citation and comment annotations, which are often species-specific) are copied over from MetaCyc to the clone. If the frame does not exist in MetaCyc, then it must have been deleted recently (otherwise it would already have been copied there), so it is deleted from the clone DB.
  - For all slot values that are copied to a clone DB and that represent frames in MetaCyc, we check to see if the corresponding frame exists in the clone DB. If not, we copy that frame to the clone DB also. The exception is if the frame is a modified protein, and the unmodified form is not present in the clone DB either, in which case we do not import either frame.

While update propagation is being performed, messages about any changes are printed to the terminal window. The user should examine these messages to make sure that the changes seem reasonable before saving the DBs. It is also good practice to save this output to a file in the $GPROOT/ecocyc/metacyc/released/kb/ directory, so that it can be examined in case changes have been lost or updates fail to be propagated (more so that we can determine if there is a bug in the propagation code rather than to recover any data, which can be retrieved from the change logs stored in Oracle). One known failing of the update propagation mechanism is that if a different change has been made to some slot of a frame in two different clone DBs (but not in MetaCyc), then the value that prevails is that from the last clone in the orgkb list (i.e. when updates are propagated upward from a clone, they may overwrite updates previously propagated upward from earlier clone DBs), and then of course that value is propagated downwards to all other clones. It is largely to avoid this problem that we

recommend that the majority of changes be performed in MetaCyc rather than in any of the clones.

The classes and slots for which update propagation is done are listed below. This set of classes and slots is currently hard-coded into the update propagation code.

– Compounds: common-name, synonyms, overview-node-shape, smiles, pka3, pka2, pka1, structure-atoms, structure-bonds, superatoms, systematic-name, n-1-name, n+1-name, molecular-weight, comment, citations, dblinks, chemical-formula, display-coords-2d, charge, cas registry-numbers, atom-chirality, atom-charges, gibbs-0, and aromatic-rings.

– EC-Reactions and Unclassified-Reactions (i.e. this excludes Transport-Reactions, which are not propagated): common-name, synonyms, requirements, right, left, substrates, dblinks, deltago, ec-number, ec-number-old, and official-ec?.

– Super-Pathways: common-name, synonyms, primaries, polymerization-links, layout-advice, net-reaction-equation, class-instance-links and disable-display.

– All Pathways except for Super-Pathways (which are covered above) and 2Comp-Pathways (which are not propagated): reaction-list, predecessors, common-name, synonyms, primaries, polymerization-links, layout-advice, net-reaction-equation, class-instance-links and disable-display.

8. DATABASE RELEASE PROCESS

The database release process is performed by BioCyc project staff at SRI. At each release, SRI curators will need to run the consistency checker tools and fix any errors identified, update the database release notes, and update general database information that is displayed to the users.

8.1. THE CONSISTENCY CHECKER

The Consistency Checker is a utility that performs various checks and corrections of the data. This program is run by SRI prior to each release, and should be run by PGDB curators periodically. The Consistency Checker builds indexes, checks citation format, checks correspondence between polypeptides and genes, changes compound names to the corresponding frame ID in some slots, updates references to compound name strings to refer to the corresponding frames, checks that physiological regulators are on the master list of regulators, checks various cross-references, checks that all reactions listed within pathways exist, checks pathway links and removes obsolete ones, checks various types of reaction information, checks for enzymatic reactions that are lacking a link to the reaction or the enzyme, checks compound structure information and removes redundant bonds, calculates sub-and superpathways, checks links to modified proteins, checks computed slot values, verifies replicon components and positions, and, finally, checks various constraints.

To run the program, open the database (e.g., EcoCyc or MetaCyc) in the Pathway Tools Navigator and select Consistency Checker from the Tools menu. The interface is divided to two main panes - automatic tasks and manual tasks. The automatic tasks find and fix the problems automatically (although the changes to the DB are not saved until the user invokes the Save command), while the manual tasks require user intervention. All tasks in each of the panels can be selected simultaneously, or the user may choose to run each task individually. When running the manual tasks, problematic objects (such as badly formed references) are printed in the right panel of the interface, and can be easily opened in the main Navigator window by clicking on them within the Consistency Checker interface. Once opened, the curator can inspect the object, analyze the cause of the problem, and repair it within the main window.

The process can be repeated as many times as required, until the errors have all been addressed. Remember to save the changes.

The Consistency Checker generates log files each time it is invoked. The log files are saved in the PGDB reports directory, and the exact path is printed within the Consistency Checker interface.

Changes performed automatically by the Consistency Checker may show up in the history record displayed by invoking the Show Changes command in the editor software. This can cause confusion while trying to understand the history of a particular object, and curators are advised to keep in mind the possibility that a series of otherwise mysterious changes to an object may be due to the Consistency Checker, rather than intentional manipulations of the object in question.

8.2. RELEASE NOTES

Updates to the EcoCyc and MetaCyc release notes are made by EcoCyc and MetaCyc curators at SRI.

8.2.1. DATABASE STATISTICS

The (kb-stats) and (new-pathways) queries are used to gather database statistics for the release notes.

After all updates to the database prior to the release have been completed, open a Lisp window and type (kb-stats :summary? t). The output generated will contain the statistics that are used in the release notes that are displayed on the BioCyc web sites.

MetaCyc release notes now include an estimate of the number of textbook pages equivalent in size to MetaCyc summaries. To include this statistic run this command instead:

(progn (kb-stats :summary? t) (comment-stats :frames (get-kb-frames)))

To generate a list of new pathways entered since the last release, type:

(new-pathways-by-date "2010-01-26" :spreadsheet? t)

Where the date is that of the freeze period of the last release. The output will provide the list of new pathways curated starting the following day.

It is advisable to save a copy of the output of each of these programs for reference.

The statistics may be gathered after the database has been frozen and saved to a file, even after curation into the MySQL database has resumed. At a Lisp prompt, type a command of this form:

(so 'META 18.5) where you substitute META and 18.5 with the appropriate PGDB name and version.

Once the file opens, you can type (kb-stats :summary? t) or ((new-pathways :spreadsheet? t) and proceed as previously described.

8.2.2. UPDATES OF THE RELEASE NOTES

The text of the release notes is found in the file ~brg/website/pgdb-release-notes-beta/[KB-NAME]/release-notes.shtml

For example, ~brg/website/pgdb-release-notes-beta/metacyc/release-notes.shtml

The file should not be edited directly. Rather, it should be checked out of cvs, updated locally, and committed.

When preparing a new version of the file, copy the format of previous sections; list the names of newly curated pathways, modified pathways, and other significant improvements. Use a horizontal rule to separate notes from different releases. MetaCyc curators use an Excel spreadsheet to automate the formatting of the modified pathways section.

8.3. UPDATES TO THE PGDB SUMMARY PAGE

The PGDB summary page is created by a part-manual, part-automated procedure during the creation of a PGDB. It includes some statistics information (e.g. the numbers of genes coding for proteins and coding for RNAs) which is not updated automatically. These values are calculated when the PGDB is created, and are cached to allow fast re-display.

All organisms in Pathway Tools databases are stored as class objects under the master class Organisms. While MetaCyc contains thousands of frames under this master class, describing the different organisms that are referenced in it, organism-specific PGDBs contain only the classes of the NCBI taxonomy that lead to the species that the PGDB describes. Under the frame describing that species (which is a class object) the PGDB contains one instance, which is the frame that describes the PGDB. In a way, a PGDB is considered an instance of the class that describes the organism. The frame-ID for that frame is the PGDB unique ID, such as ECOLI for E. coli. In its Genome slot, that frame will point to the genetic elements.

Several items shown in the PGDB summary page can be edited using the PGDB Info Editor (right-click on the PGDB name and select Edit → PGDB Info Editor). These include the list of authors, the copyright information, and citation information. This information should be checked and updated, if necessary, prior to each release.

In addition, the cached statistics can be updated by running the “Recompute database statistics” command from the consistency Checker.

9. REFERENCES

[1 Karp PD, Paley S, Krieger CJ and Zhang P (2004). An evidence ontology for use in pathway/genome databases. In R. Altman and T. Klein, editors, Proc Pacific Symposium on Biocomputing, pages 190–201, Singapore, World Scientific.

[2] Karp PD, Paley SM, Krummenacker M, Latendresse M, Dale JM, Lee TJ, Kaipa P, Gilham F, Spaulding A, Popescu L, Altman T, Paulsen I, Keseler IM and Caspi R (2010). Pathway Tools version 13.0: Integrated Software for Pathway/Genome Informatics and Systems Biology. Briefings in Bioinformatics 11:40-79.

[3] Caspi R, Altman T, Dreher K, Fulcher CA, Keseler IM, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Ong Q, Paley S, Pujar A, Shearer A, Subhraveti P, Travers M, Weerasinghe D, Zhang P and Karp PD (2012). The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Research 40(1):D742-D753

[4] Karp PD, Riley M, Saier M, Paulsen IT, Paley S and Pellegrini-Toole A. (2002) The EcoCyc database. Nuc Acids Res, 30(1):56-8.

[5] Keseler IM, Collado-Vides J, Santos-Zavaleta A, Peralta-Gil M, Gama-Castro S, Muniz-Rascado L, Bonavides-Martinez C, Paley S, Krummenacker M, Altman T, Kaipa P, Spaulding A, Pacheco J, Latendresse M, Fulcher C, Sarker M, Shearer AG, Mackie A, Paulsen I, Gunsalus RP and Karp PD (2011) EcoCyc: a comprehensive database of Escherichia coli biology. Nucleic Acids Research 39:D583-590.