Bioinformatics, 32(7), 2016, 1074–1079
doi: 10.1093/bioinformatics/btv702
Advance Access Publication Date: 1 December 2015
Original Paper

Databases and ontologies

Representation and inference of cellular architecture for metabolic reconstruction and modeling

Suzanne Paley, Markus Krummenacker and Peter D. Karp*

Bioinformatics Research Group, SRI International, Menlo Park, CA 94025, USA

*To whom correspondence should be addressed.

Associate Editor: Janet Kelso

Received on 12 June 2015; revised on 16 October 2015; accepted on 25 November 2015

Abstract

Motivation: Metabolic modeling depends on accurately representing the cellular locations of enzyme-catalyzed and transport reactions. We sought to develop a representation of cellular compartmentation that would accurately capture cellular location information. We further sought a representation that would support automated inference of the cellular compartments present in newly sequenced organisms to speed model development, and that would enable representing the cellular compartments present in multiple cell types within a multicellular organism.
Results: We define the cellular architecture of a unicellular organism, or of a cell type from a multicellular organism, as the collection of cellular components it contains plus the topological relationships among those components. We developed a tool for inferring cellular architectures across many domains of life and extended our Cell Component Ontology to enable representation of the inferred architectures. We provide software for visualizing cellular architectures to verify their correctness and software for editing cellular architectures to modify or correct them. We also developed a representation that records the cellular compartment assignments of reactions with minimal duplication of information.
Availability and implementation: The Cell Component Ontology is freely available. The Pathway Tools software is freely available for academic research and is available for a fee for commercial use.
Contact: pkarp@ai.sri.com
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Accurate representation of the compartmentation of the cell is an important issue in metabolic modeling. In our quest to streamline the processes of developing metabolic models, and of exchanging models between different research groups, a number of issues related to cellular compartmentation arise. What is the essential information about cellular compartmentation that should be captured, and how can it be effectively represented? How can we automatically infer the cellular compartments of an organism when creating a new metabolic model? When that automatic inference fails, how can we facilitate manual adjustment of compartmentation by a user?

A Cell Component Ontology (CCO) should be able to represent the universal set of cellular components from which all known cells are built, as well as the particular cellular components from which a specific type of cell is built. For multicellular organisms, the CCO should be able to represent, within one database, the cellular components of the multiple cell types present in that organism. Cell types in multicellular organisms sometimes have different cellular architectures, e.g. red blood cells lack nuclei. CCO should capture the information needed to support the inference of the cellular architecture of different organisms, such as by describing the typical organization of cellular components in different types of organisms (e.g. the cellular

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

© The Author 2015. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com

architecture of a typical plant versus a typical bacterium) in a way that can be instantiated in a particular organism. Finally, CCO should enable inferences about the spatial organization of cellular compartments (e.g. if a transporter is assigned to a given membrane within a cell, a program should be able to infer that the transporter transports metabolites from the cellular space surrounding that membrane to the cellular space surrounded by that membrane).

Accurate metabolic models must define the different sets of reactions that occur within different cellular compartments of a given cell type, and the exchange reactions that translocate metabolites across cellular membranes. Consider the problem of modeling a large community of interacting organisms by combining metabolic models created by different research groups for individual organisms. For example, imagine that we wanted to combine a human intestinal epithelial cell metabolic model derived from RECON2 Thiele *et al.* (2013) with models of bacteria from the gut microbiota Shoaie and Nielsen (2014). If the individual models do not identify their common extracellular space using a single shared identifier, it will not be possible for the metabolites secreted by one organism to be taken up by another organism. Furthermore, for consistency and understandability it is desirable for different models to use the same controlled vocabulary terms to refer to the same cellular components, such as the bacterial inner membrane. Standardization of the cellular compartments used in individual metabolic models would clearly facilitate our ability to combine organism models created by different research groups in a plug-and-play fashion. Currently, the models produced by different research groups use a wide array of *ad hoc* and incompatible compartment identifiers, e.g. 'golgi apparatus' [Recon 2 Thiele *et al.* (2013)] versus 'golgi' [Yeast 7 Heavner *et al.* (2013)], 'extra cellular space' (Recon 2) versus 'extracellular' (Yeast 7), versus 'extra-organism' [MSB Chang *et al.* (2011)].

An overview of our approach to these problems is as follows. We have updated our CCO Zhang *et al.* (2005) http://bioinformatics.ai.sri.com/CCO/to enable the concurrent description of multiple cell types within a single database and to permit development of metabolic reconstructions that span multiple (e.g. human) cell types. The CCO describes 'surrounds' relationships among compartments (e.g. the cytosol surrounds the mitochondrion). Proteins, metabolites and reactions within a Pathway/Genome Database (PGDB) can be annotated with one or more CCO terms to define their cellular location(s).

Further, each CCO cellular component is labeled with the National Center for Biotechnology Information (NCBI) Taxonomy identifiers of the taxonomic groups in which that component is expected to appear. This information enables inferring the cellular components present in an organism, given its NCBI Taxonomy identifier, by the metabolic reconstruction module (PathoLogic) of our Pathway Tools [overview: Karp *et al.* (2015a); summary of recent enhancements: Karp *et al.* (2015b)] software. We refer to the set of cellular components present in a unicellular organism, or in a type of cell from a multicellular organism, plus the topological relationships among those cellular components, as the cellular architecture. When a user finds the cellular architecture of an organism to be in error, they can use the new Cellular Architecture Editor within Pathway Tools to revise the cellular architecture. The MetaFlux Karp *et al.* (2015b) metabolic modeling module within Pathway Tools uses the CCO terms attached to metabolic and transport reactions to assign those reactions to appropriate cellular compartments within a metabolic model.

We have defined a mapping between CCO terms and Gene Ontology (GO) Gene Ontology Consortium (2015) terms to maximize the standardization of CCO while enabling it to be extended independently of GO when necessary.

2 Results

2.1 Cell component ontology

CCO is implemented as a set of class terms and a set of instance terms. Each leaf CCO class term describes a general type of cellular component (e.g. class term CCO-RGH-ER-MEM describes the class of all rough endoplasmic reticulum membranes found in all cell types). Parent CCO class terms define an IS-A hierarchy above these leaf class terms (e.g. class term CCO-MEMBRANE defines the class of all membranes and is an IS-A parent of CCO-RGH-ER-MEM).

CCO instance terms describe specific cellular components found in specific cell types. For example, instance term CCI-RGH-ER-MEM-1 (an instance of CCO-RGH-ER-MEM) might describe the rough endoplasmic reticulum membrane found in a human red blood cell in the HumanCyc database Romero *et al.* (2004), and instance CCI-RGH-ER-MEM-2 might describe the rough endoplasmic reticulum membrane found in a human hepatocyte in the HumanCyc database.

Table 1 lists the slots (database fields) defined for CCO terms. These slots are used to provide metadata that define the meanings of CCO terms, and to define spatial relationships among CCO terms. Based on these spatial relationships, we have written software that creates cartoons depicting cellular architectures such as that shown for *Escherichia coli* in Figure 1. These cartoons enable users to quickly validate the correctness of the cellular architecture defined by a collection of CCO terms (that is, to validate that the cellular architecture contains the correct set of organelles and their

Table 1. Slots used in the CCO

Slot	Type	Meaning
ID	A	The unique identifier of the CCO term.
COMMON-NAME	A	The English name of the CCO term.
DEFINITION	A	The definition of a CCO term.
DEFINITION-REFERENCE	A	The source from which the definition of a CCO term was derived.
GOID	A	The GO identifier of the GO term corresponding to this CCO term.
IN-CELL-ARCHITECTURE	A	The organism or cell type in the current database to which this cell component instance belongs. This slot is only populated for CCO instances (see Section 2.3), and each instance will link either to a single cell type object or to the object that describes the current organism.
SENSU	A	Restricts this CCO term to be used in organisms subsumed by one or more NCBI Taxonomy IDs listed in this slot.
SURROUNDS	R	Lists one or more CCO terms for cell components that are directly physically surrounded by this component.
SURROUNDED-BY	R	Lists one or more CCO terms for cell components that directly physically surround this cellular component.
COMPONENTS	R	Lists one or more CCO terms for cell components that are direct parts of this cellular component.
COMPONENT-OF	R	Lists one or more CCO terms for cell components of which this cellular component is a direct part.

Slots of type 'A' define attributes; slots of type 'R' define relationships between a given cellular component and one or more other cellular components.

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

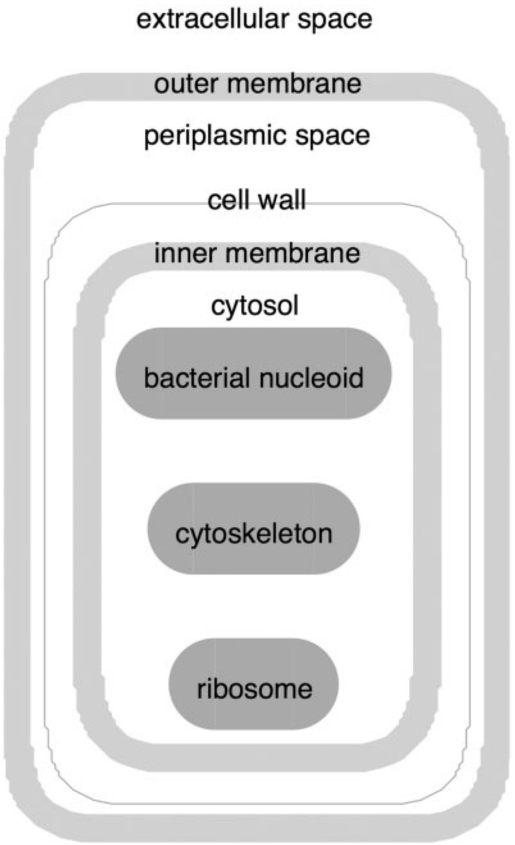


Fig. 1. Cellular architecture cartoon generated by our software from the CCO description of the *E.coli* cellular architecture

components organized with the proper surrounds relationships). Supplementary Figure S1 contains generated cartoons for a number of other organisms (a Gram-positive bacterium, a cyanobacterium, a plant, a trypanosome, a human liver cell and a human red blood cell) to demonstrate the range of cellular architectures that can be captured with CCO. Note that the software that generates the cartoons currently omits cell projections, such as flagella, because our SURROUNDS/SURROUNDED-BY relationships do not capture the geometry of these components relative to other cell components.

CCO contains seven primitive types of entities, each described by a top-level class within the ontology:

- Membranes
- Spaces
- Organelles
- Sub-organelle compartments (e.g. golgi cisterna)
- Cell envelopes
- Super-components that consist of a space plus other entities contained within the space (e.g. the cytoplasm or a spore)
- A cell-surface matrix such as a cell wall

2.2 Cellular compartments of reactions and metabolites

Here, we describe how Pathway Tools represents the compartments of reactions and metabolites to facilitate metabolic modeling. Supplementary File S1 provides a tutorial on how to interactively assign reaction compartments within Pathway Tools.

One key design goal is avoiding unnecessary duplication of information. Thus, we create a single object for a given metabolite or a given reaction in a PGDB, even if biologically, a metabolite or a reaction occurs in more than one compartment. Reactions can be annotated with information about what compartments they reside in. Because metabolites are connected to the reactions, we infer that if metabolite M is a reactant or product of reaction R, then M is present in all of the locations in which R is present.

Another goal is reusing reactions curated in MetaCyc Caspi *et al.* (2014) (our universal reaction database). When PathoLogic infers Karp *et al.* (2011) the reactions present in an organism, it imports those reactions from MetaCyc into the new PGDB created for that organism. This implies that a reaction must have an abstract description of how it fits topologically into a cell's architecture, without relying on hard-coding potentially exotic compartments that might not even exist in the new target PGDB. Instead, PathoLogic should be able to assign the correct, concrete compartments for the reaction in the target PGDB. For example, consider the reaction for the choline:H+ symporter, which transports choline. That transport event might take place across different membranes in different organisms. The MetaCyc transport reaction must describe in an abstract way whether choline is on the inside or outside relative to the membrane that is crossed, both on the left and right sides of the reaction equation. We assume that the inside is surrounded by the outside. In this example, if Pathologic imports the reaction into a Gram-positive bacterium, then the inside will be equated with the cytosol and the outside with the extracellular space. However, if this reaction is imported into a Gram-negative bacterium, then the inside will again be equated with the cytosol, whereas the outside will now be equated with the periplasmic space.

We define two types of reactions:

- **S-reactions** (S = Same) are reactions in which all reactants and products (jointly: substrates) of the reaction reside in the same compartment. This is the most common type of metabolic reaction and corresponds to enzyme-catalyzed reactions. The substrates of S-reactions are stored in slots called LEFT (reactants) and RIGHT (products), respectively.
- **T-reactions** (T = Transport) are reactions whose substrates are translocated from one compartment to another, crossing a membrane. Their substrates reside in more than one compartment, most commonly, two compartments. Some T-reactions involve translocation but no chemical modification, whereas other T-reactions chemically modify their substrates (e.g. phosphorylation). T-reaction substrates are also stored in the LEFT and RIGHT slots.

The coefficients of substrates in both S-reactions and T-reactions are stored by using a database structure called an annotation, which is provided by the Ocelot database Karp *et al.* (1999) system used by Pathway Tools. An annotation is a labeled list of data values that can be attached to a slot value. For example, if the reactant H_2O had a coefficient of 2 in the LEFT slot of a given reaction, we would attach the Annotation 2 under the label COEFFICIENT to the value H_2O in the LEFT slot of that reaction.

The locations of the substrates within S-reactions and T-reactions are assigned by using the following approaches.

S-reaction location assignment: Because the substrates of all S-reactions are located in the same compartment, by specifying the compartment of the reaction, we specify the locations of all of its substrates. By default, if no location information is asserted for them, S-reactions are presumed to be located in the cytosol (specifically, CCO-CYTOSOL). If an S-reaction occurs in a non-default compartment, or in several compartments, then the RXN-LOCATIONS slot of that reaction stores the CCO identifier of each such compartment (which must all be children of CCO-SPACE). The metabolites in the reaction's LEFT and RIGHT slots are

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

interpreted as existing together in every compartment listed in the RXN-LOCATIONS slot.

The following example first shows the equation of an S-reaction located in the periplasm; it then shows the internals of the object representing the S-reaction. That object has the slots LEFT and RIGHT, containing metabolite objects as their values, and slot RXN-LOCATIONS, containing a CCO object as its value.

```
  An example of an S-reaction in the periplasmic
space:
  2-phospho-D-glycerate[periplasmic space] +
  H2O[periplasmic space]
                 ->
  D-glycerate[periplasmic space] +
  phosphate[periplasmic space]
  Reaction PHOSPHOGLYCERATE-PHOSPHATASE-RXN
    LEFT: 2-PG, WATER
    RIGHT: GLYCERATE, Pi
    RXN-LOCATIONS: CCO-PERI-BAC
```

T-reaction location assignment: We specify the locations of T-reaction substrates by specifying the membrane(s) that the T-reaction spans, and by specifying the locations of each substrate relative to each membrane. T-reactions do not have a default membrane assignment, because for cells with multiple organelle membranes, which membrane should be the default is unclear. The membrane(s) spanned by a T-reaction are stored in the RXNS-LOCATIONS slot of the reaction. The relative locations of each T-reaction substrate are specified by using abstract location annotations called CCO-IN and CCO-OUT. This information is combined programmatically to derive specific locations for every substrate in a T-reaction.

For each membrane spanned by a T-reaction, the CCO-IN and CCO-OUT annotations on a substrate specify whether the substrate resides inside that membrane, or outside the membrane. These abstract compartment annotations on the metabolites carry the label COMPARTMENT. The RXN-LOCATIONS slot further specifies, for each membrane, bindings of CCO-IN and CCO-OUT to actual compartment spaces.

An example of a transport reaction in the periplasmic membrane, also called the inner membrane, is shown as follows. In this example, the transport reaction bridges the membrane CCO-PM-BAC-NEG (the periplasmic membrane of a Gram-negative bacterium), and the pair of bindings declares that CCO-IN maps to CCO-CYTOSOL, and CCO-OUT maps to CCO-PERI-BAC. The representation for this example depicts the annotations of a slot value on a new line underneath the value itself, with three dashes preceding the label of an annotation, followed by the value of the annotation. For example, the value CCO-PM-BAC-NEG is annotated with the value CCO-CYTOSOL under the label CCO-IN.

```
  An example of a T-reaction in the inner membrane:

  D-glycerate[periplasmic space] -> D-glycerate
[cytosol]
  Reaction TRANS-RXN0-523
    LEFT:
      GLYCERATE
      —COMPARTMENT: CCO-OUT
    RIGHT:
      GLYCERATE
      —COMPARTMENT: CCO-IN
    RXN-LOCATIONS:
      CCO-PM-BAC-NEG
      —CCO-IN: CCO-CYTOSOL —CCO-OUT: CCO-PERI-BAC
```

Here, we provide an example of how the location of a substrate within this T-reaction is computed by the following Pathway Tools Application Program Interface (API) call:

```
  get-reactant-location (TRANS-RXN0-523,
                         GLYCERATE,
                         CCO-PM-BAC-NEG)
```

The API first queries whether glycerate has CCO-IN or CCO-OUT as its abstract compartment. Its annotation for the label COMPARTMENT is CCO-OUT. Next we look up the specific location by examining the annotation that has the label that is the same as the abstract compartment we found at the substrate, namely CCO-OUT. This finally leads us to CCO-PERI-BAC as the specific location of GLYCERATE on the LEFT side.

Whenever a reaction is transferred by PathoLogic from MetaCyc to a new PGDB as part of reaction inference, all values in the RXN-LOCATIONS slot are filtered away (i.e. *not* copied). This prevents inapplicable compartments from being introduced into other PGDBs. Instead, inferring appropriate compartments will be the responsibility of Pathologic; in future work, PathoLogic will infer reaction locations based on information such as protein signal sequences.

2.3 Inference of cellular architecture

Our convention is for every PGDB to contain all CCO classes to define the universe of cellular components—even for cellular components not present in that organism. However, instances of those classes are defined for only those cellular components that are present in the organism. A PathoLogic module infers the instances to create (and thus, the cellular architecture of the organism) based on the taxonomic class to which the organism belongs. Curators can edit the cellular architecture to make any needed corrections, as described in the next section.

The inference process uses information present in the CCO classes. The SENSU slot of each CCO class term links (via NCBI taxonomy id) to those taxa that can potentially contain that component. The linked taxa can be very general—many components, such as the cytosol, the plasma membrane and the extracellular space, will be applicable to all organisms. Or they can be quite specific (for example the CCO-THY-CYA class, which describes the cyanobacterial thylakoid, is specific for Cyanobacteria). Some class terms have several values for this slot (for example, the CCO-PERI-BAC class, which refers to the periplasm in Gram-negative bacteria, links to all of the bacterial taxa that have an outer membrane enclosing a periplasm). Thus, given the taxonomic group of an organism, we can automatically infer which CCO terms are expected to belong to that organism. We do not actually create instances for every one of these class terms, just for the 'leaf' class terms (i.e. those that do not have a more specific class). For example, CCO-PLASMA-MEM refers to the plasma membrane and is present in all organisms, but its subclass CCO-PM-BAC-NEG specifically represents the plasma membrane in Gram-negative bacteria (analogous subclasses exist for other taxa—taxon-specific subclasses are necessary because different taxa can have different values for the SURROUNDS or COMPONENTS relationships). For a Gram-negative bacterium, we would create an instance only of the latter, more specific class.

After the software has created a set of instance terms for an organism, it copies the SURROUNDS and COMPONENTS

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

relationships from the parent classes to the corresponding instances. Sometimes multiple instances for a given class term must be created and correctly linked. For example, CCO contains class terms for several different kinds of vesicles (e.g. endocytic vesicles, synaptic vesicles, etc.). The components of a vesicle are the vesicle membrane and the vesicle lumen. For each kind of vesicle present in an organism, a different vesicle membrane instance and vesicle lumen instance must be created, and each membrane instance must surround the correct lumen instance.

After the set of instances and their interrelationships have been created for an organism, we should have a consistent picture of the cellular components for that organism that can be used to, say, draw a cartoon such as in Figure 1 or to determine the appropriate membranes and spaces for transport reactions.

2.4 Cellular Architecture Editor

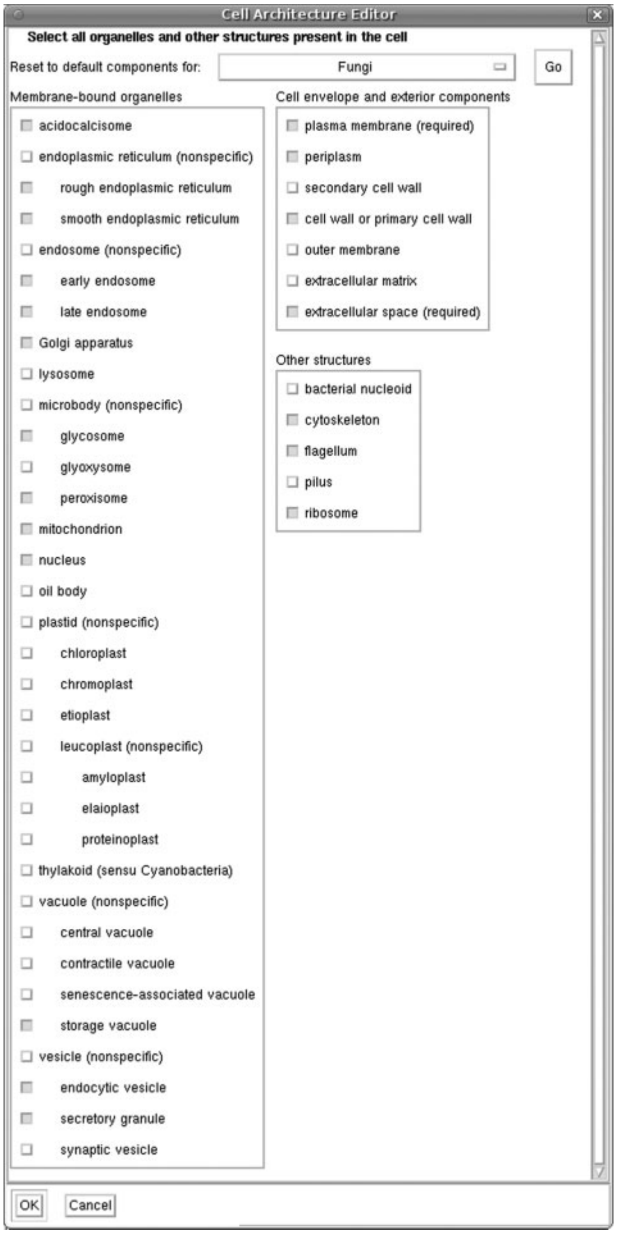
Although we have invested considerable effort to ensure that the SENSU values described earlier are as accurate and precise as possible, some cases will exist in which the set of components inferred based on taxonomy will not be entirely correct. For example, many but not all Euglenids have chloroplasts, but because the chloroplast-related terms all link to the Euglenid taxon (among others), those terms will be inferred by default for all Euglenids, even the non-photosynthetic ones. Thus, we also provide a Cellular Architecture Editor, so that a curator can correct the set of cellular components belonging to an organism. This editor can also be used to define the architecture of different cell types in a multicellular organism—for example, although both are human, the set of components in a human erythrocyte is very different than that in a human hepatocyte.

The Cellular Architecture Editor, shown in Figure 2, enables the user to select which cellular components are present or absent in a particular organism or cell type. It does not list every possible CCO class term, however. Terms are organized into three categories: (i) cell envelope and exterior components, (ii) membrane-bound organelles and (iii) other. If the curator specifies that a particular organelle is present, then all of its component terms are automatically included also. For example, if an organism includes the mitochondrion, then it automatically also includes the terms for the mitochondrial membranes, lumen, intermembrane space, etc. The editor can be preset with the components present in a generic plant, animal, fungal, Gram-negative or Gram-positive bacterial cell, or with the set of inferred components for the current organism. This ability enables easily starting with a reasonable base set of components and then just adding or subtracting whatever is necessary.

2.5 Mapping between CCO and GO

Most CCO terms contain links to the corresponding term in the GO cellular component ontology. However, several differences exist between CCO and GO.

GO contains a great many more terms than CCO because it chooses to represent cellular components at a higher level of detail than we do, often all the way down to individual protein complexes. For our purposes, we have chosen to keep CCO focused on large-scale cellular compartments and structures. Because these more specific GO terms are generally parts of components whose terms do have mappings to CCO, we have no trouble determining the appropriate correspondence.



Cell Architecture Editor

Select all organelles and other structures present in the cell

Reset to default components for: Fungi

Membrane-bound organelles

▨ acidocalcisome
□ endoplasmic reticulum (nonspecific)
▨   rough endoplasmic reticulum
▨   smooth endoplasmic reticulum
□ endosome (nonspecific)
▨   early endosome
▨   late endosome
▨ Golgi apparatus
□ lysosome
□ microbody (nonspecific)
▨   glycosome
□   glyoxysome
▨   peroxisome
▨ mitochondrion
▨ nucleus
□ oil body
□ plastid (nonspecific)
□   chloroplast
□   chromoplast
□   etioplast
□   leucoplast (nonspecific)
□   amyloplast
□   elaioplast
□   proteinoplast
□ thylakoid (sensu Cyanobacteria)
□ vacuole (nonspecific)
□   central vacuole
□   contractile vacuole
□   senescence-associated vacuole
▨   storage vacuole
□ vesicle (nonspecific)
▨   endocytic vesicle
▨   secretory granule
□   synaptic vesicle

Cell envelope and exterior components

▨ plasma membrane (required)
▨ periplasm
□ secondary cell wall
▨ cell wall or primary cell wall
□ outer membrane
□ extracellular matrix
▨ extracellular space (required)

Other structures

□ bacterial nucleoid
▨ cytoskeleton
▨ flagellum
□ pilus
▨ ribosome

Fig. 2. The cellular architecture editor

Both GO and CCO represent two kinds of relationships between terms: IS-A relationships and PART-OF relationships. In addition, CCO represents a third kind of relationship not present in GO, the SURROUNDED-BY relationship, which, as we have seen earlier, lets us infer the topology of a cell. As a consequence, CCO sometimes has several sensu-specific terms that map to a single GO term. For example, in addition to the CCO term CCO-PLASMA-MEM, the cell plasma membrane, which corresponds to the GO term GO:0005886, CCO has additional child terms for the plasma membrane in plants, animals, fungi and Gram-negative and Gram-positive bacteria. These additional terms exist in CCO to enable us to represent the different SURROUNDED-BY relationships in different taxa. For example, the Gram-negative plasma membrane is surrounded by the bacterial periplasm, whereas the Gram-positive plasma membrane is surrounded by the extracellular space. Although these sensu-specific CCO terms do not directly link to any GO term, generating a mapping between the two vocabularies if the organism taxon is known is trivial.

A small number of CCO terms lack any mapping to GO. These terms are mostly very high-level generic concepts, such as

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016

CCO-SPACE (any 3D extent, typically bound by and/or surrounding a membrane, such as the cytosol, extracellular-space and various organelle lumens) and CCO-SUBORG-CMPT (the parent of all suborganelle compartments). Because we do not expect any proteins or reaction substrates to be assigned directly to any of these terms, the lack of corresponding GO terms should not cause any problems. We also have a term CCO-UNKNOWN-SPACE to help us explicitly represent uncertainty, which GO chooses not to do. Finally, a few generic CCO instance terms—such as CCO-SIDE-1, CCO-SIDE-2 and CCO-BOUNDARY—are placeholders that enable us to define generic transport reactions in MetaCyc without reference to any specific membrane or compartments. When these reactions are brought into an organism-specific database, the generic terms must be mapped to specific cellular components within that cell's architecture.

The earlier exceptions notwithstanding, the mapping between CCO and GO means that outputting the cellular location of any protein or reaction substrate by using either vocabulary is straightforward. In fact, in our internal representation, we make use of both. Protein locations are encoded by using GO terms (to use the higher level of detail contained in GO, and to facilitate exchange with other databases that use GO), and the corresponding CCO terms are computed. The locations of reaction substrates are encoded by using CCO, because for reactions, the topology defined by the SURROUNDED-BY relationships is paramount.

3 Discussion

The representations described herein are used to describe cellular architectures in all of the organisms listed in Figure 1 and in Supplementary Figure S1. These representations are used to describe reaction locations in EcoCyc Keseler *et al.* (2013) and HumanCyc and are used in the metabolic model generated from EcoCyc Weaver *et al.* (2014).

Currently, metabolic models produced by different research groups use varying *ad hoc* schemes for labeling cellular compartments. This approach slows and complicates the utilization of both individual metabolic models (where the meanings of compartment names may be unclear) and attempts to combine multiple individual models into larger community models—if different models use different labels for the same compartments, the modeling software will not realize what compartments are shared by different organisms. We recommend CCO as a standard for describing the architectures of cells for use in metabolic models, and Pathway Tools uses CCO terms when exporting its metabolic models to Systems Biology Markup Language (SBML) Hucka *et al.* (2003).

4 Conclusions

We define the cellular architecture of a unicellular organism, or a cell type from a multicellular organism, as the collection of cellular components it contains plus the topological relationships among those components. We have extended our CCO to enable inferring the cellular architectures of cell types across many domains of life. One extension annotates the CCO classes that represent cellular components with taxonomic identifiers that indicate the taxa in which those cellular components are found. Another extension enables creating multiple instances of those classes, where each instance describes a component present in a different cell type. The Pathway Tools software contains tools to visualize cellular architectures to enable their verification, and editing to modify or correct cellular architectures.

We also developed a representation that records the cellular compartment assignments of a reaction within the reaction object, thus avoiding duplication of reactions.

Funding

This work was supported by award numbers R01GM075742 and GM092729 from the National Institute of General Medical Sciences. The content of this article is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health.

Conflict of Interest: none declared.

References

Caspi,R. *et al.* (2014) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res.*, **42**, D459–D471.

Chang,R.L. *et al.* (2011) Metabolic network reconstruction of chlamydomonas offers insight into light-driven algal metabolism. *Mol. Syst. Biol.*, **7**, 518.

Gene Ontology Consortium. (2015) Gene Ontology Consortium: going forward. *Nucleic Acids Res.*, **43**, D1049–D1056.

Heavner,B.D. *et al.* (2013) Version 6 of the consensus yeast metabolic network refines biochemical coverage and improves model performance. *Database*, **2013**, bat059.

Hucka,M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Karp,P. *et al.* (1999) A collaborative environment for authoring large knowledge bases. *J. Intell. Inf. Syst.*, **13**, 155–194.

Karp, P. *et al.* (2015a) Pathway Tools version 19.0: Integrated software for pathway/genome informatics and systems biology, arXiv.org, 1–79. arxiv.org/abs/1510.03964.

Karp,P. *et al.* (2015b) Pathway Tools version 19.0 update: Software for pathway/genome informatics and systems biology. *Brief. Bioinform.*, doi: 10.1093/bib/bbv079.

Karp,P.D. *et al.* (2011) The Pathway Tools pathway prediction algorithm. *Stand. Genomic Sci.*, **5**, 424–429.

Keseler,I.M. *et al.* (2013) EcoCyc: fusing model organism databases with systems biology. *Nucleic Acids Res.*, **41**, D605–D612.

Romero,P. *et al.* (2004) Computational prediction of human metabolic pathways from the complete human genome. *Genome Biol.*, **6**, 1–17.

Shoaie,S. and Nielsen,J. (2014) Elucidating the interactions between the human gut microbiota and its host through metabolic modeling. *Front. Genet.*, **5**, 86.

Thiele,I. *et al.* (2013) A community-driven global reconstruction of human metabolism. *Nat. Biotechnol.*, **31**, 419–425.

Weaver,D. *et al.* (2014) A genome-scale metabolic flux model of *E. coli* K–12 derived from the EcoCyc database. *BMC Syst. Biol.*, **8**, 79.

Zhang,P. *et al.* (2005) MetaCyc and AraCyc. metabolic pathway databases for plant research. *Plant Physiol.*, **138**, 27–37.

Downloaded from http://bioinformatics.oxfordjournals.org/ at University of California, Los Angeles on August 30, 2016