The minimum information requested in the annotation of SBML models (Hucka et al., 2003) were identified solely by free-text, human-readable, name attributes, often resulting in equivalent components being named differently in different models. As naming conventions are non-standard, it is impossible to definitively match these components computationally, and the process of model merging then requires human input to resolve ambiguities. Providing MIRIAM-compliant annotations allows a component to be unambiguously identified by associating it with one or more terms from publicly available databases such as ChEBI (Degtyarenko et al., 2008) or KEGG (Kanehisa et al., 2000) (Fig. 1).

2 FEATURES

The species elements in Figure 1 are both annotated with MIRIAM-compliant terms. libSBML (Bornstein et al., 2008) provides the facility for reading a given SBML element’s annotation and hence could be used to determine that species1 and species2 are annotated with ChEBI term CHEBI:4167 and KEGG Compound C00031, respectively. From this, it may be concluded that the compounds represented by these species are different. However, manual inspection of the database references in ChEBI and KEGG show that both species are annotated with references that share the same chemical structure, and hence are equivalent.

Performing such a comparison computationally is beyond the scope of libSBML. To do so, the annotations must be ‘dereferenced’ by querying the two databases via their web service interfaces. This task is complicated particularly because each of the web services has non-standard interfaces.

The libAnnotationSBML library creates a unified framework for supporting MIRIAM-compliant annotations by wrapping these divergent web services into a Java API, allowing each web service to be queried in a consistent manner. The library itself can act as a layer on top of the libSBML API.

The library is built dynamically by querying the MIRIAM web service (Laibe and Le Novère, 2007), which provides a collection of data types that are recommended for use in model annotation.

The web service provides details of each of these data types to existing data resources.

1 Manchester Centre for Integrative Systems Biology, Manchester Interdisciplinary Biocentre, University of Manchester, Manchester M1 7DN, UK and 2 Virginia Bioinformatics Institute, Virginia Tech, Washington St. 0477, Blacksburg, VA 24061, USA

© 2009 The Author(s)

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Fig. 2. Class diagram showing public methods of OntologyTerm and specialized subclasses ChebiTerm, UniProtTerm and KeggReactionTerm.

including names, URNs and physical URLs to resources. From this, a collection of Ontology objects are instantiated, one for each data type supported specified in MIRIAM.

Individual OntologyTerms objects are built up from an Ontology object and a unique identifier. Once instantiated, the OntologyTerm provides a number of methods, specified in Figure 2. The implementation of these methods is performed by mapping the calls to an appropriate call to the data type's web service, where such a web service exists.

The OntologyTerm class can be extended to provide methods specific to the SBML element that is being described. For example, a metabolite species element annotated with a Chebi term will return a ChebiTerm object, providing a method for accessing the chemical formula of the metabolite. Similarly, a KEGG Reaction annotation will return a KeggReactionTerm object, providing methods for accessing reactants and products, each returned as OntologyTerms themselves.

Applying libAnnotationSBML to the SBML in Figure 1 will associate an OntologyTerm with each of the species. Calling getTerm() on these CHEBI and KEGG OntologyTerm objects returns 'd-glucopyranose' and 'd-glucose', respectively. Considering the initial problem of comparing SBML components, this provides an example of why names cannot be used reliably to perform this task. A more reliable approach is to exploit the fact that many data resources cross-reference one another. For example, entries in the ChEBI database can provide details of the equivalent term in KEGG, and vice versa. The OntologyTerm class supports this by implementing a getXrefs() method which returns cross-references themselves as OntologyTerms, along with a predicate, defined in libSBML, that indicates the relationship between them. When an OntologyTerm references an equivalent entity in a different database, the predicate libsbmlConstants.BQB_ENCODES is returned. In the case of a genomic database entry cross referencing an entity in a proteomic database, libsbmlConstants.BQB_ENCODES is useful. Using this method, it can be determined computationally that the ChEBI and KEGG terms cross-reference one another, and hence species1 and species2 can be unambiguously determined to represent equivalent entities.

The libAnnotationSBML library facilitates the rapid development of tools to manipulate SBML annotation terms. The library can be used to add annotation to unannotated SBML models, using a similar approach to semanticSBML (Schulz et al., 2006). libAnnotationSBML can annotate both metabolites and proteins, exploiting the search facility that exists in both the ChEBI and UniProt web services (The UniProt Consortium, 2008).

The focus of libAnnotationSBML is to develop tools to manipulate already annotated models. An example of such a tool is the SbnlSynonymExtractor, which takes annotated SBML as input, and returns a mapping of all species terms to their name synonyms, harvested from ChEBI, KEGG or UniProt. Another tool, the SbnlReactionBalancer, determines whether the reactions specified within an SBML file are elementally balanced by querying the ChEBI web service to retrieve chemical formulae of reaction participants.

libAnnotationSBML was used extensively in the development of a genome-scale model of yeast metabolism, the first model of this scale in which all compartments, metabolites, enzymes and complexes are unambiguously defined using MIRIAM-compliant annotations (Herrgård et al., 2008).

3 IMPLEMENTATION AND DISTRIBUTION

The API is written in Java 1.5 and is dependent upon libSBML v3. It is supported in Linux, Windows and MacOS X and is distributed as source code and associated build files under the open source Academic Free Licence v3.0 from http://mcisb.sf.net/ along with other tools described in this manuscript.

ACKNOWLEDGEMENTS

The authors thank the BBSRC and EPSRC for financial support of this work was an integral part.

Conflict of Interest: none declared.

REFERENCES

Biemann,B.J. et al. (2008) LibSBML: an API Library for SBML. Bioinformatics, 24, 880–881.

Dytyverenko,K. et al. (2008) KEGG: a database and ontology for chemical entities of biological interest. Nucleic Acids Res., 36, D344–D350.

Herrgård,M. et al. (2008) A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. Nature Biotechnol., 26, 1155–1160.

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.

Kanehisa,M. et al. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res., 28, 27–30.

Kell,D.B. et al. (2008) LibSBML: an API Library for SBML. Bioinformatics, 24, 880–881.

Laibe,C. and Le Novere,N. (2007) MIRIAM resources: tools to generate and resolve robust cross-references in Systems Biology. BMC Syst. Biol., 1, 58.

Le Novere,N. et al. (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nature Biotechnol., 23, 1509–1515.

Schulz,M. et al. (2006) SBML-energy, a system for combining biochemical network models. Genome Inform. Ser., 17, 62–71.

The UniProt Consortium (2008) The universal protein resource (UniProt). Nucleic Acids Res., 36, D190–D195.