SBML Level 3 Package: Flux Balance Constraints ('fbc')

Brett G. Olivier
b.g.olivier@vu.nl
Systems Bioinformatics
VU University Amsterdam
Amstedam, NH, The Netherlands

Frank T. Bergmann
fbergmann@caltech.edu
Computing and Mathematical Sciences
California Institute of Technology
Pasadena, CA, US

August 14, 2012

Version 1 (Draft)

α-release version 0.2 (August 14, 2012)

The latest release, past releases, and other materials related to this specification are available at
http://sbml.org/Documents/Specifications/Fbc

This release of the specification is available at
http://sbml.org/Documents/Specifications/Fbc_Level_1_Version_1

Contents

1 Introduction and motivation

Constraint based modelling is a widely used methodology used to analyse and study biological networks on both a small and whole organism (genome) scale. Typically these models are underdetermined and constraint based methods (e.g. linear, quadratic optimization) are used to optimise specific model properties. This is assumed to occur under a defined set of constraints (e.g. stoichiometric, metabolic) and bounds (e.g. thermodynamic, experimental and environmental) on the values that the solution fluxes can obtain.

Perhaps the most well known (and widely used) analysis method is Flux Balance Analysis (FBA; Orth et al., 2010) which is performed on Genome Scale Reconstructions (GSR's; Oberhardt et al., 2009). Using FBA a target flux is optimized (e.g. maximising a flux to biomass or minimising ATP production) while other fluxes can be bounded to simulate a selected growth environment or specific metabolic state.

As constraint based models are generally underdetermined, i.e. few or none of the kinetic rate equations and related parameters are known, it is crucial that a model definition includes the ability to define optimisation parameters such as objective functions, flux bounds and constraints ... currently this is not possible in the Systems Biology Markup Language (SBML) Level 2 or Level 3 core specification (Hucka et al., 2011, 2003).

The question of how to encode constraint based (a.k.a. 'FBA') models in SBML is not new. However, advances in the methods used to construct GSR scale models and the wider adoption of constraint based modelling in biotechnological/medical applications have led to a rapid increase in both the number of models being constructed and the tools used to analyse them.

Faced with such growth, both in number and diversity, the need for a standardised data format for the definition, exchange and annotation of constraint based models has become critical. As the core model components (e.g. species, reactions, stoichiometry) can already be efficiently described in SBML (with its significant community, software and tool support) the Flux Balance Constraints Package aims to extend SBML core by adding the elements necessary to describe current and future constraint based models.

1.1 Proposal corresponding to this package specification

This specification for Flux Balance Constraints in SBML Level 3 Version 1 is based on the proposal by the same authors, located at the following URL:

`http://sbml.org/Community/Wiki/SBML_Level_3_Proposals/Flux_Balance_Constraints_Proposal_(2012)`

The tracking number in the SBML issue tracking system (SBML Team, 2010) for Flux Balance Constraints package activities is 3154219. The version of the proposal used as the starting point for this specification is the version of March 2012. Previous versions of the current proposal are:

Version 3 (March 2012)

`http://sbml.org/Community/Wiki/SBML_Level_3_Proposals/Flux_Balance_Constraints_Proposal_(2012)`

Version 2 (March 2011)

`http://sbml.org/Community/Wiki/SBML_Level_3_Proposals/Flux_Constraints_Proposal`

Version 1 (February 2010)

`http://precedings.nature.com/documents/4236/version/1`

Details of earlier independent proposals are described in Section 2.

1.2 Package dependecies

The Flux Balance Constraints package has no dependencies on other SBML Level 3 packages. It is also designed with the goal of being able to work seamlessly with other SBML Level 3 packages. For example the new elements are entirely encapsulated in their own ListOf classes and any extensions to existing SBML classes are defined as optional.

1.3 Document conventions

Following the precedent set by the SBML Level 3 Core specification document, we use UML 1.0 (Unified Modeling Language; Eriksson and Penker 1998; Oestereich 1999) class diagram notation to define the constructs provided by this package. We also use color in the diagrams to carry additional information for the benefit of those viewing the document on media that can display color. The following are the colors we use and what they represent:

- *Black*: Items colored black in the UML diagrams are components taken unchanged from their definition in the SBML Level 3 Core specification document.
- *Green*: Items colored green are components that exist in SBML Level 3 Core, but are extended by this package. Class boxes are also drawn with dashed lines to further distinguish them.
- *Blue*: Items colored blue are new components introduced in this package specification. They have no equivalent in the SBML Level 3 Core specification.

We also use the following typographical conventions to distinguish the names of objects and data types from other entities; these conventions are identical to the conventions used in the SBML Level 3 Core specification document:

AbstractClass: Abstract classes are classes that are never instantiated directly, but rather serve as parents of other object classes. Their names begin with a capital letter and they are printed in a slanted, bold, sans-serif typeface. In electronic document formats, the class names defined within this document are also hyperlinked to their definitions; clicking on these items will, given appropriate software, switch the view to the section in this document containing the definition of that class. (However, for classes that are unchanged from their definitions in SBML Level 3 Core, the class names are not hyperlinked because they are not defined within this document.)

Class: Names of ordinary (concrete) classes begin with a capital letter and are printed in an upright, bold, sans-serif typeface. In electronic document formats, the class names are also hyperlinked to their definitions in this specification document. (However, as in the previous case, class names are not hyperlinked if they are for classes that are unchanged from their definitions in the SBML Level 3 Core specification.)

`SomeThing`, `otherThing`: Attributes of classes, data type names, literal XML, and generally all tokens *other* than SBML UML class names, are printed in an upright typewriter typeface. Primitive types defined by SBML begin with a capital letter; SBML also makes use of primitive types defined by XML Schema 1.0 (Biron and Malhotra, 2000; Fallside, 2000; Thompson et al., 2000), but unfortunately, XML Schema does not follow any capitalization convention and primitive types drawn from the XML Schema language may or may not start with a capital letter.

For other matters involving the use of UML and XML, we follow the conventions used in the SBML Level 3 Core specification document.

2 Background

2.1 Problems with current SBML approaches

While there is currently no official way of encoding constraint based models in SBML L2 there have been pragmatic approaches used by a variety of groups and applications. Arguably the best and most widely used format is that used by the COBRA toolbox (Becker et al., 2007) where the metabolic network is well defined using SBML **Reaction** and **Species** classes. However, flux bounds and reactions that take part in the objective function are defined as **LocalParameter** classes and (implicitly) rely on all tools using the same naming convention. Similarly, reaction annotations are generally stored as key-value pairs in HTML **Notes** elements which has led to different groups routinely using different keys describing describing the same thing. While a step in the right direction this format is not suitable for implementation in SBML Level 3.

2.2 Past work on this problem or similar topics

The problem of describing and annotating 'FBA' models in SBML has been raised at various times in the past few years. In this regard there are two known putative proposals one by Karthik Raman and the other by the Church Laboratory. As far as we are aware these proposals never developed beyond their initial presentation at SBML forum/hackathons. In 2009 the discussion was reopened at the SBML Forum held in Stanford and has subsequently developed into the current active package proposal and this document (see Section 1).

Brett Olivier (2009) SBML Level 3 FBA package discussion

http://sbml.org/images/4/4a/Olivier_sbml_forum_2009_09_04.pdf

Karthik Raman (2005) Flux annotations in SBML

http://sbml.org/images/d/d9/Raman-flux-annotations.pdf

Church laboratory (pre 2005) Metabolic flux model annotations

http://sbml.org/Community/Wiki/Old_known_SBML_annotations_list

3 Proposed syntax and semantics

In this section, we define the syntax and semantics of the Flux Balance Constraints package for SBML Level 3 Version 1. We expound on the various data types and constructs defined in this package, then in Section 4 on page 10, we provide complete examples of using the constructs in example SBML models.

3.1 Namespace URI and other declarations necessary for using this package

Every SBML Level 3 package is identified uniquely by an XML namespace URI. For an SBML document to be able to use a given SBML Level 3 package, it must declare the use of that package by referencing its URI. The following is the namespace URI for this version of the Flux Balance Constraints package for SBML Level 3 Version 1:

"`http://www.sbml.org/sbml/level3/version1/fbc/version1`"

In addition, SBML documents using a given package must indicate whether understanding the package is required for complete mathematical interpretation of a model, or whether the package is optional. This is done using the attribute `required` on the <sbml> element in the SBML document. For the Flux Balance Constraints package, the value of this attribute must be set to "`true`".

The following fragment illustrates the beginning of a typical SBML model using SBML Level 3 Version 1 and this version of the Flux Balance Constraints package:

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" level="3" version="1"
      xmlns:fbc="http://www.sbml.org/sbml/level3/version1/fbc/version1" fbc:required="true">
```

3.2 Primitive data types

Section 3.1 of the SBML Level 3 Version 1 Core specification defines a number of primitive data types and also uses a number of XML Schema 1.0 data types (Biron and Malhotra, 2000). More specifically we make use of `integer`, `double`, `string`, `SIdRef` and `enum`. In addition we make use of two new primitives `FbcSId` and `FbcSIdRef`, see Figure 1 for the interrelation between these entities.

3.2.1 Type `FbcSId`

The type `FbcSId` is derived from `SId` (SBML Level 3 Version 1 Core specification Section 3.1.7) and has identical syntax. The `FbcSId` type is used as the data type for the identifiers of **FluxBound** (Section 3.5) and **Objective** (Section 3.6) classes. By using a separate identifier type we differentiate them from others defined in the **SBML** model and thus ensuring data encapsulation. In addition the **Objective** class `FbcSId` provides an identifier to the **Objective** which is set as active. The equality of `FbcSId` values is determined by an exact character sequence match and therefore comparisons of these identifiers must be performed in a case-sensitive manner.

3.2.2 Type `FbcSIdRef`

Type `FbcSIdRef` is used for all attributes that refer to identifiers of type `FbcSId`. This type is derived from `FbcSId` with the restriction that the value of an attribute having type `FbcSIdRef` must match the value of a `FbcSId` attribute in the current model. In the FBC package the **ListOfObjectives** has an attribute of this type that is used to refer to an instance of the **Objective** class.

3.3 The extended Model class

The **SBML Model** class is extended with the addition of two children, i.e. a `listOfFluxBounds` and a `listOfObjectives`.

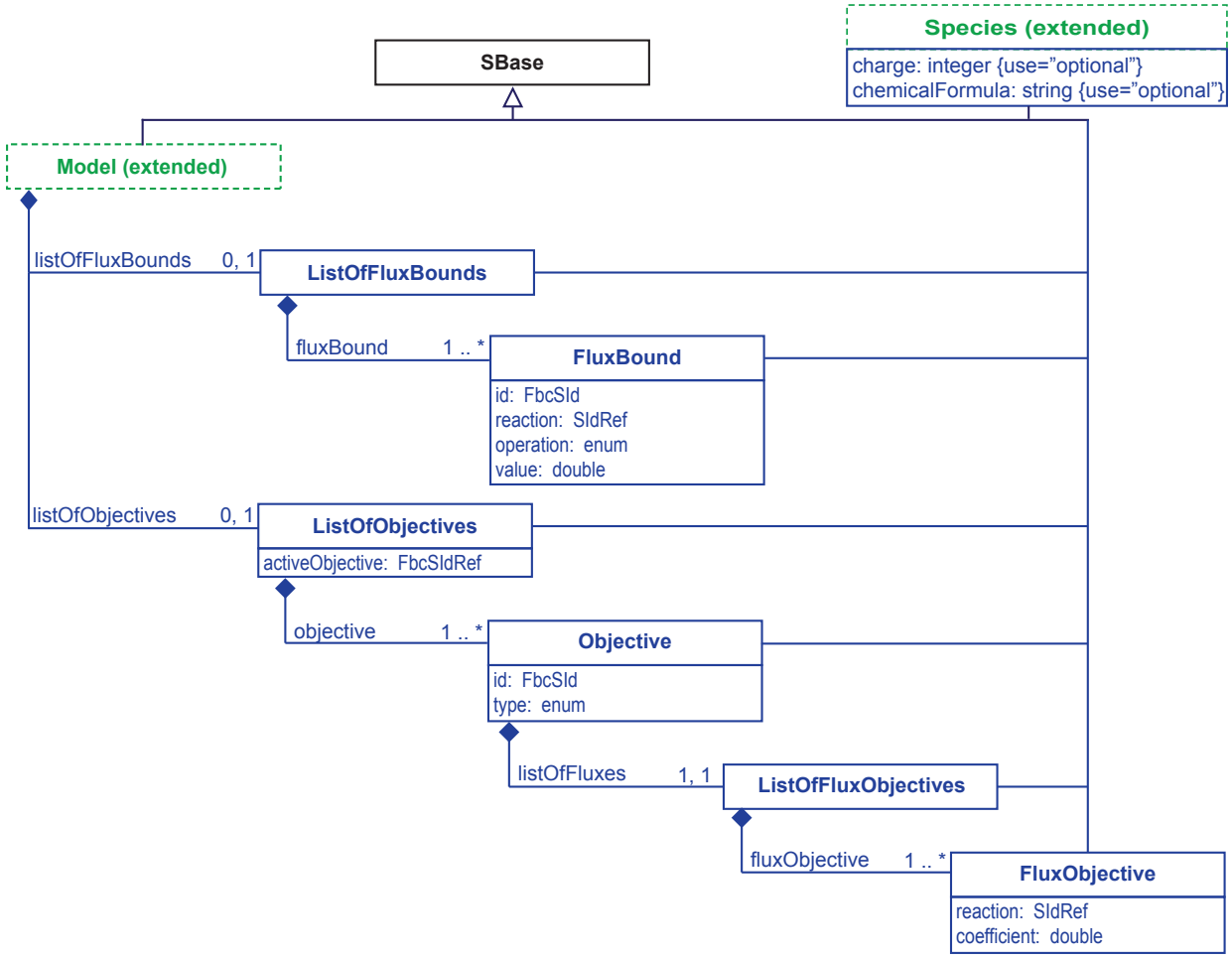

Figure 1: *A UML representation of the Flux Balance Constraints package classes. See Section 1.3* for conventions related to this figure.

3.3.1 The lists of objectives and flux bounds

As shown in Figure 1 the **ListOfFluxBounds** and **ListOfObjectives** are derived from ***SBase*** and inherit `metaid` and `sboTerm`, as well as the subcomponents for **Annotation** and **Notes**. Both of these list are required to contain one or more elements when defined, however, the lists themselves are optional. Unlike most other **SBML ListOf__** classes, **ListOfObjectives** introduces an additional required attribute `activeObjective`.

The `activeObjective` *attribute*

This attribute contains a "`value`" of type `FbcSIdRef` that can only refer to an existing **Objective** (FbcSId). This required attribute exists so that when multiple **Objective** 's are included in a single model, the model will always be well described i.e. there is a single, primary objective function which closes the solution space.

3.4 The extended Species class

The FBC package extends the **SBML Species** class with the addition of two attributes:

- an optional attribute `charge` which contains an `integer` referring the the **Species**' charge (as defined in **SBML** Level 2)

- an optional attribute `chemicalFormula` containing a `string` that represents the **Species**' elemental composition.

The* `chemicalFormula` *attribute

While there are many ways of referring to an elemental composition the purpose of the `chemicalFormula` attribute is to allow reaction balancing and validation which is particularly important in constraint based models. To this end it is recommended that the format of `chemicalFormula` should follow the Hill system (or notation). Here the number of carbon atoms in a molecule is indicated first, followed by the number of hydrogen atoms and then the number of all other chemical elements in alphabetical order. When the formula contains no carbon; all elements, including hydrogen, are listed alphabetically Hill (1900, 2012).

3.5 The FBC FluxBound class

FluxBound is a new FBC class derived from **SBML *SBase*** that inherits `metaid` and `sboTerm`, as well as the subcomponents for **Annotation** and **Notes**. The purpose of this class is to hold a single (in)equality that provides the maximum or minimum value that a reaction flux can obtain at steady state. It is is relatively straight forward and implements four attributes.

- An `id` attribute that contains an `FbcSId`,
- a `reaction` attribute that takes an `SIdRef` and can take **SBML Reaction** `SId` as a value,
- an `operation` attribute of type `enum` that can take a limited set of boolean operators (see text for details),
- `value` an attribute that takes a `double` value representing the bound. This may include $\pm\infty$ encoded as the value

The* `operation` *attribute

The `operation` attribute represents a mathematical (in)equality of the form <`reaction`> <`operator`> <`value`> e.g.

$$R_5 >= 0$$
$$R_5 < \infty$$
$$R_7 = 1.0$$

An enumerated type that can take one of the following values:

$$\begin{aligned} <= &\mapsto \text{“}\texttt{lessEqual}\text{”} \\ >= &\mapsto \text{“}\texttt{greaterEqual}\text{”} \\ < &\mapsto \text{“}\texttt{less}\text{”} \\ > &\mapsto \text{“}\texttt{greater}\text{”} \\ = &\mapsto \text{“}\texttt{equal}\text{”} \\ \text{undefined} &\mapsto \text{“}\texttt{unknown}\text{”} \end{aligned}$$

3.6 The FBC Objective class

As shown in Figure 1 the FBC **Objective** class is derived from **SBML *SBase*** and inherits `metaid` and `sboTerm`, as well as the subcomponents for **Annotation** and **Notes**. An integral component in a complete description of a steady-state model is the so-called 'objective function' which generally consist of a linear combination of model variables (fluxes) and a sense (direction). In the FBC package this concept is succinctly captured in the **Objective** class containing three required attributes:

- `id` an attribute that can only contain an `FbcSId`,
- `type` an `enum` (see below),
- `listOfFluxes` which contains a **ListOfFluxes** .

The* `type` *attribute

The `type` attribute contains an `enum` which represents the sense of the optimality constraint and can take one of three values:

$$\begin{aligned} maximize &\mapsto \text{“}\texttt{maximize}\text{”} \\ minimize &\mapsto \text{“}\texttt{minimize}\text{”} \\ \text{undefined} &\mapsto \text{“}\texttt{unknown}\text{”} \end{aligned}$$

The* `listOfFluxes` *attribute

The **ListOfFluxes** is derived from and functions like a typical **SBML ListOf__** class with the proviso that it cannot be empty and must contain one or more `fluxObjective` attributes of type **FluxObjective** (see Section 3.7).

3.7 The FBC FluxObjective class

As shown in Figure 1 the FBC **FluxObjective** class is derived from **SBML *SBase*** and inherits `metaid` and `sboTerm`, as well as the subcomponents for **Annotation** and **Notes**.

The **FluxObjective** class is a relatively simple container for a model variable weighted by a signed coefficient:

- `id` an attribute that contains an `SId` that is restricted to refer only to an **SBML Reaction**,
- `coefficient` a `double`.

4 Examples

This section contains a variety of examples of SBML Level 3 Version 1 documents employing the Flux Balance Constraints package.

4.1 FBC syntax examples

Encoding the FluxBound

As described in Section 3.5 the flux bound represents a mathematical (in)equality of the form <reaction> <operator> <value>. In SBML Level 3 Version 1 with FBC this is encoded as:

```
<fbc:listOfFluxBounds>
 <fbc:fluxBound fbc:id="R1b" fbc:reaction="R1" fbc:operation="greaterEqual" fbc:value="1.2"/>
 <fbc:fluxBound fbc:id="R2b" fbc:reaction="R2" fbc:operation="lessEqual" fbc:value="-1.2"/>
 <fbc:fluxBound fbc:id="R3b" fbc:reaction="R3" fbc:operation="greaterEqual" fbc:value="-INF"/>
 <fbc:fluxBound fbc:id="R4b" fbc:reaction="R4" fbc:operation="lessEqual" fbc:value="INF"/>
 <fbc:fluxBound fbc:id="R5b" fbc:reaction="R5" fbc:operation="equal" fbc:value="1"/>
 <fbc:fluxBound fbc:id="R6b" fbc:reaction="R6" fbc:operation="greater" fbc:value="-1"/>
 <fbc:fluxBound fbc:id="R7b" fbc:reaction="R7" fbc:operation="less" fbc:value="1"/>
</fbc:listOfFluxBounds>
```

This example illustrates two things: the encoding of ∞ and that care should be used when selecting inequalities such as or . While mathematically there is a difference, this difference is only practically relevant when working with rational arithmetic (solvers).

Encoding the Objective

The FBC allows for the definition of multiple 'objective functions' with one being designated as active (see Section 3.6) the following example illustrates this:

```
<fbc:listOfObjectives fbc:activeObjective="obj1">
 <fbc:objective fbc:id="obj1" fbc:type="maximize">
  <fbc:listOfFluxes>
   <fbc:fluxObjective fbc:reaction="R101" fbc:coefficient="1"/>
  </fbc:listOfFluxes>
 </fbc:objective>
 <fbc:objective fbc:id="obj2" fbc:type="minimize">
  <fbc:listOfFluxes>
   <fbc:fluxObjective fbc:reaction="R102" fbc:coefficient="-2.5"/>
   <fbc:fluxObjective fbc:reaction="R103" fbc:coefficient="1"/>
  </fbc:listOfFluxes>
 </fbc:objective>
</fbc:listOfObjectives>
```

Note how both **Objective** instances differ in `type` and each contains different set of **FluxObjectives**.

4.2 Examples contrasting the current SBML L2 encoding with L3 and FBC

These examples contrast some elements of an existing model, iJR904 from the BiGG Database encoded in the COBRA format Becker et al. (2007); Reed et al. (2003); Schellenberger et al. (2010) that have been translated into SBML Level 3 Version 1 using the CBMPy implementation of the FBC package Olivier (2012); Olivier et al. (2005) and libSBML experimental ver. 5.6.0 Bornstein et al. (2008).

Objective function definition

Old style SBML Level 2 objective

```
<reaction id="R_BiomassEcoli" name="BiomassEcoli" reversible="false">
<kineticLaw>
<math xmlns="http://www.w3.org/1998/Math/MathML">
<ci>FLUX_VALUE</ci>
</math>
<listOfParameters>
<parameter id="LOWER_BOUND" value="0" units="mmol_per_gDW_per_hr"/>
<parameter id="UPPER_BOUND" value="999999" units="mmol_per_gDW_per_hr"/>
<parameter id="OBJECTIVE_COEFFICIENT" value="1" />
<parameter id="FLUX_VALUE" value="0" units="mmol_per_gDW_per_hr"/>
</listOfParameters>
</kineticLaw>
</reaction>
```

New SBML Level 3 style objective

```
<fbc:listOfObjectives fbc:activeObjective="obj1">
 <fbc:objective fbc:id="obj1" fbc:type="maximize">
  <fbc:listOfFluxes>
   <fbc:fluxObjective fbc:reaction="R_BiomassEcoli" fbc:coefficient="1"/>
  </fbc:listOfFluxes>
 </fbc:objective>
</fbc:listOfObjectives>
```

Species definition

It is particularly useful to contrast the differences in **Species** definition as used in constraint based, genome scale models.

Old SBML Level 2 style species

To begin with we let's examine the **SBML** Level 2 Version 1 species definition used by the BiGG database and COBRA Becker et al. (2007); Schellenberger et al. (2010). Note how the `name` attribute is overloaded with the chemical formula.

```
<species id="M_atp_c" name="ATP_C10H12N5O13P3"
    compartment="Cytosol" charge="-4" />
```

An alternate SBML Level 2 style annotation

A newer variation of the above, probably necessitated by the discontinuation of the `charge` attribute in **SBML** and libSBML

```
<species id="M_atp_c" name="ATP" compartment="c">
  <notes>
    <body xmlns="http://www.w3.org/1999/xhtml">
      <p>FORMULA: C10H12N5O13P3</p>
      <p>CHARGE: -4</p>
    </body>
  </notes>
</species>
```

New SBML Level 3 style species

Hopefully, with the adoption of **SBML** FBC these species properties can be unified into a common format.

```
<species metaid="meta_M_atp_c␣" id="M_atp_c" name="ATP"
    compartment="Cytosol" boundaryCondition="false"
    fbc:charge="-4" fbc:chemicalFormula="C10H12N5O13P3"
    initialConcentration="0" hasOnlySubstanceUnits="false"
/>
```

Reaction definition and flux bounds

Old SBML Level 2 style reaction

```
<reaction id="R_GTHS" name="glutathione␣synthetase" reversible="false">
<notes>
<html:p>Abbreviation: R_GTHS</html:p>
<html:p>Synonyms: _0</html:p>
<html:p>EC Number: 6.3.2.3</html:p>
<html:p>SUBSYSTEM: Cofactor and Prosthetic Group Biosynthesis</html:p>
<html:p>Equation: [c] : atp + glucys + gly --&gt; adp + gthrd + h + pi</html:p>
<html:p>Confidence Level: 0</html:p>
<html:p>NCD</html:p>
<html:p>genes:</html:p>
```

```
<html:p>LOCUS:b2947#ABBREVIATION:gshB#ECNUMBERS:6.3.2.3#</html:p>
<html:p>proteins:</html:p>
<html:p>NAME:glutathione synthase#ABBREVIATION:GshB#</html:p>
<html:p>GENE ASSOCIATION: (b2947)</html:p>
</notes>
<listOfReactants>
<speciesReference species="M_atp_c" stoichiometry="1"/>
<speciesReference species="M_glucys_c" stoichiometry="1"/>
<speciesReference species="M_gly_c" stoichiometry="1"/>
</listOfReactants>
<listOfProducts>
<speciesReference species="M_adp_c" stoichiometry="1"/>
<speciesReference species="M_gthrd_c" stoichiometry="1"/>
<speciesReference species="M_h_c" stoichiometry="1"/>
<speciesReference species="M_pi_c" stoichiometry="1"/>
</listOfProducts>
<kineticLaw>
<math xmlns="http://www.w3.org/1998/Math/MathML">
<ci>FLUX_VALUE</ci>
</math>
<listOfParameters>
<parameter id="LOWER_BOUND" value="0" units="mmol_per_gDW_per_hr"/>
<parameter id="UPPER_BOUND" value="999999" units="mmol_per_gDW_per_hr"/>
<parameter id="OBJECTIVE_COEFFICIENT" value="0" />
<parameter id="FLUX_VALUE" value="0" units="mmol_per_gDW_per_hr"/>
</listOfParameters>
</kineticLaw>
</reaction>
```

New SBML Level 3 style reaction and flux bound

Please note that in order to maintain all the annotation encoded in the **SBML** L2 reaction `notes` an additional (tool specific) annotation is introduced i.e. 'KeyValueData'. This should be considered a transitional step until a generally accepted annotation system is adopted by the constraint based modelling community. Nevertheless, where (unambiguously) possible, the L2 annotation has been converted into a MIRIAM compliant form e.g. the 'EC number'.

```
<reaction metaid="meta_R_GTHS" id="R_GTHS" name="glutathione␣synthetase" reversible="false">
 <annotation>
  <listOfKeyValueData xmlns="http://pysces.sourceforge.net/KeyValueData">
   <data id="subsystem" type="string" value="Cofactor␣and␣Prosthetic␣Group␣Biosynthesis"/>
   <data id="name" type="string" value="glutathione␣synthase#ABBREVIATION:GshB#"/>
   <data id="gene_association" type="string" value="(b2947)"/>
   <data id="equation" type="string" value="[c]␣:␣atp␣+␣glucys␣+␣gly␣--&gt;␣adp␣+␣gthrd␣+␣h␣+␣pi"/>
   <data id="genes" type="string"/>
   <data id="proteins" type="string"/>
   <data id="locus" type="string" value="b2947#ABBREVIATION:gshB#ECNUMBERS:6.3.2.3#"/>
   <data id="abbreviation" type="string" value="R_GTHS"/>
   <data id="synonyms" type="string" value="_0"/>
   <data id="confidence_level" type="string" value="0"/>
  </listOfKeyValueData>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"
        xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
        xmlns:bqmodel="http://biomodels.net/model-qualifiers/">
   <rdf:Description rdf:about="#meta_R_GTHS">
    <bqbiol:is>
     <rdf:Bag>
      <rdf:li rdf:resource="http://identifiers.org/ec-code/6.3.2.3"/>
     </rdf:Bag>
    </bqbiol:is>
   </rdf:Description>
  </rdf:RDF>
 </annotation>
</reaction>
```

```
<fbc:listOfFluxBounds>
 <fbc:fluxBound fbc:id="R_GTHS_lower_bnd" fbc:reaction="R_GTHS" fbc:operation="greaterEqual"
    fbc:value="0"/>
 <fbc:fluxBound fbc:id="R_GTHS_upper_bnd" fbc:reaction="R_GTHS" fbc:operation="lessEqual"
    fbc:value="999999"/>
</fbc:listOfFluxBounds>
```

5 Best practices

In this section, we recommend a number of practices for using and interpreting various constructs in the Flux Balance Constraints package. These recommendations are non-normative, but we advocate them strongly; ignoring them will not render a model invalid, but may reduce interoperability between software and models.

Acknowledgments

We thank Herbert Sauro (University of Washington, Washington, USA), Neil Swainston (University of Manchester, Manchester, UK), Keiran Smallbone (University of Manchester), and Michael Hucka (Caltech, California, USA) for help developing this Flux Balance Constraints package for SBML Level 3. We also thank everyone who contributed ideas and discussion over the years.

In addition we would like to thank the authors of the ‘contrib’ package whose text was used both as an inspiration (and occasionally literally).

References

Becker, S. A., Feist, A. M., Mo, M. L., Hannum, G., Palsson, B. O., and Herrgard, M. J. (2007). Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox. *Nat. Protocols*, 2(3):727–738. 1750-2799 10.1038/nprot.2007.99 10.1038/nprot.2007.99.

Biron, P. V. and Malhotra, A. (2000). XML Schema part 2: Datatypes (W3C candidate recommendation 24 October 2000). Available via the World Wide Web at http://www.w3.org/TR/xmlschema-2/.

Bornstein, B. J., Keating, S. M., Jouraku, A., and Hucka, M. (2008). Libsbml: an api library for sbml. *Bioinformatics*, 24(6):880–881.

Eriksson, H.-E. and Penker, M. (1998). *UML Toolkit*. John Wiley & Sons, New York.

Fallside, D. C. (2000). XML Schema part 0: Primer (W3C candidate recommendation 24 October 2000). Available via the World Wide Web at http://www.w3.org/TR/xmlschema-0/.

Hill, E. A. (1900). On a system of indexing chemical literature; adopted by the classification division of the u. s. patent office. *Journal of the American Chemical Society*, 22(8):478–494.

Hill, E. A. (2012). Wikipedia: The hill system. Available via the World Wide Web at http://en.wikipedia.org/wiki/Hill_system.

Hucka, M., Bergmann, F. T., Hoops, S., Keating, S. M., Sahle, S., and Wilkinson, D. J. (2011). Available via the World Wide Web at http://sbml.org/Documents/Specifications.

Hucka, M., Finney, A., Sauro, H., Bolouri, H., Doyle, J., Kitano, H., Arkin, A., Bornstein, B., Bray, D., Cornish-Bowden, A., Cuellar, A., Dronov, S., Gilles, E., Ginkel, M., Gor, V., Goryanin, I., Hedley, W., Hodgman, T., Hofmeyr, J., Hunter, P., Juty, N., Kasberger, J., Kremling, A., Kummer, U., Novère, N. L., Loew, L., Lucio, D., Mendes, P., Minch, E., Mjolsness, E., Nakayama, Y., Nelson, M., Nielsen, P., Sakurada, T., Schaff, J., Shapiro, B., shimizu, T., Spence, H., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., and Wang, J. (2003). The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–31.

Oberhardt, M. A., Palsson, B. Ø., and Papin, J. A. (2009). Applications of genome-scale metabolic reconstructions. *Molecular Systems Biology*, 5:320.

Oestereich, B. (1999). *Developing Software with UML: Object-Oriented Analysis and Design in Practice*. Addison-Wesley.

Olivier, B. G. (2012). CBMPy: constraint based modelling in python. Available via the World Wide Web at http://pysces.sf.net/cbm.

Olivier, B. G., Rohwer, J. M., and Hofmeyr, J.-H. S. (2005). Modelling cellular systems with pysces. *Bioinformatics*, 21(4):560–561.

Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010). What is flux balance analysis? *Nature Biotechnology*, 28(3):245–248.

Reed, J. L., Vo, T. D., Schilling, C. H., and Palsson, B. O. (2003). An expanded genome-scale model of escherichia coli k-12 (ijr904 gsm/gpr). *Genome Biol*, 4(9):R54.

SBML Team (2010). The SBML issue tracker. Available via the World Wide Web at http://sbml.org/issue-tracker.

Schellenberger, J., Park, J., Conrad, T., and Palsson, B. (2010). Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinformatics*, 11(1):213.

Thompson, H. S., Beech, D., Maloney, M., and Mendelsohn, N. (2000). XML Schema part 1: Structures (W3C candidate recommendation 24 October 2000). Available online via the World Wide Web at the address http://www.w3.org/TR/xmlschema-1/.