Systems Biology Markup Language (SBML) Level 3 Proposal: Distributions within MathML

Colin Gillespie[1*], Darren Wilkinson[1], Richard Boys[1],
Carole Proctor[2], Daryl Shanley[2], Tom Kirkwood[2]

July 4, 2005

[1] School of Mathematics and Statistics University of Newcastle, Newcastle upon Tyne NE4 6BE.
[2] SCMS - Gerontology, University of Newcastle, Henry Wellcome Laboratory for Biogerontology Research, Newcastle General Hospital, Newcastle upon Tyne NE4 6BE.

*email: c.gillespie@ncl.ac.uk

1. Introduction

This document describes proposed features for inclusion in Systems Biology Markup Language (SBML) Level 3. These features allow both continuous and discrete distributions to be included within an SBML model. These features are derived from the needs of the BASIS project (http://www.basis.ncl.ac.uk).

This document simply presents some features which could be included in SBML Level 3. We expect that a number changes will need to be made to this document after review from the community.

The definitions found in the Appendices are taken from the Gnu Scientific Library at http://www.gnu.org/software/gsl/manual/gsl-ref_toc.html.

An example model implanting some of the features described in this document can be found in Appendix III.

2. Random Numbers

MathML would now contain the new expressions:

normalRandom: a continuous Normal random variable with mean μ and standard deviation σ;

gammaRandom: a continuous Gamma random variable with parameters a and b;

exponentialRandom: a continuous Exponential random variable with mean μ;

uniformRandom: a continuous Uniform random variable with parameters a and b;

uniformDiscreteRandom: a discrete Uniform random variable with parameters a and b;

poissonRandom: a discrete Poisson random variable with mean λ.

A fuller explanation of these functions is given in Appendix I.

Since SBML Level 2 uses the MathML csymbol element to denote certain built-in mathematical entities, i.e. delay and time, then it seems natural that these random variables would be called in the same manner.

The following examples demonstrate how the random numbers would be called. The XML fragment below codes for a uniform(0,1) random variable

```
<xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <csymbol encoding="text"
         definitionURL="http://www.sbml.org/sbml/symbols/uniformRandom">
      uniformRandom
    </csymbol>
    <cn>0</cn>
    <cn>1</cn>
  </apply>
</math>
```

This fragment codes for a Normal variable with mean μ and standard deviation σ; the values of μ and σ are set somewhere else, e.g. as a parameter

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <csymbol encoding="text"
          definitionURL="http://www.sbml.org/sbml/symbols/normalRandom">
      normalRandom
    </csymbol>
    <ci>mu</ci>
    <ci>sigma</ci>
  </apply>
</math>
```

3. Density Functions

To complement the above random number generators, we include their respective probability density/mass functions. These new expressions are: normalDensity, uniformDensity, gammaDensity, exponentialDensity, uniformDiscreteMass, and poissionMass. The complete definitions can be found in Appendix II. This fragment of xml codes for the density of the Normal distribution at $x = 2$ with mean 0 and standard deviation 1.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <csymbol encoding="text"
         definitionURL="http://www.sbml.org/sbml/symbols/normalDensity">
      normalDensity
    </csymbol>
    <cn>2</cn>
```

```
    <cn>0</cn>
    <cn>1</cn>
  </apply>
</math>
```

This piece of xml should return $p(2)$ where p is defined in Appendix II.

4. Other Possibilities

4..1 Other Distributions

We have (intentionally) included only a small subset of distributions with the hope that everyone will adopt them. Are there any other distributions that should be included? For other possibilities see the random number section for the gnu scientific library at http://www.gnu.org/software/gsl/manual /gsl-ref_toc.html.

4..2 Cumulative Distribution Function

We decided against including the cumulative distribution function mainly to make the proposal as simple as possible. Should it be included?

4..3 Truncated Distributions

This proposal has omitted truncated distributions. Are they wanted/need?

Appendix I - Random Variables

Random: *double* **normalRandom**(*double* μ, *double* σ)

This function returns a Gaussian random variate, with mean μ and standard deviation σ. The probability distribution for Gaussian random variates is,

$$p(x)dx = \frac{\exp(-(x-\mu)^2/(2\sigma^2))}{\sqrt{2\pi\sigma^2}}dx$$

for x in the range $-\infty$ to $+\infty$.

Random: *double* **gammaRandom**(*double* a, *double* b)

This function returns a random variate from the gamma distribution. The distribution function is,

$$p(x)dx = \frac{1}{\Gamma(a)b^a}x^{a-1}e^{-x/b}dx$$

for $x > 0$.

Random: *double* **uniformRandom**(*double* a, *double* b)

This function returns a random variate from the flat (uniform) distribution from a to b. The distribution is,

$$p(x)dx = \frac{1}{b-a}dx$$

if $a \leq x < b$ and 0 otherwise.

Random: *double* **exponentialRandom** (*double* μ)

This function returns a random variate from the exponential distribution with mean μ. The distribution is,

$$px(x)dx = \frac{e^{-x/\mu}}{\mu}dx$$

for $x > 0$.

Random: *int* **uniformDiscreteRandom**(*integer* a, *integer* b)

This function returns a random variate from the discrete uniform distribution from a to b. The distribution is defined as,

$$p(k) = \frac{1}{b-a}$$

for $k = a, \ldots, b-1$ and 0 otherwise.

Random: *int* **poissonRandom**(*double* λ)

This function returns a random integer from the Poisson distribution with mean λ. The probability distribution for Poisson variates is,

$$p(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

for $k \geq 0$ and $\lambda > 0$.

Appendix II - Density and Mass Functions

Function: *double* gammaDensity(*double* x, *double* a, *double* b)

This function computes the probability density $p(x)$ at x for a gamma distribution with parameters a and b, using the formula given in Appendix I.

Function: *double* uniformDensity(*double* x, *double* a, *double* b)

This function computes the probability density $p(x)$ at x for a uniform distribution from a to b, using the formula given in Appendix I.

Function: *double* normalDensity (*double* x, *double* μ, *double* σ) This function computes the probability density $p(x)$ at x for a Gaussian distribution with mean μ and standard deviation σ, using the formula given in Appendix I.

Function: *double* exponentialDensity (*double* x, *double* μ)

This function computes the probability density $p(x)$ at x for an exponential distribution with mean μ, using the formula given in Appendix I.

Function: *double* poissionMass(*int* k, *double* λ)

This function computes the probability $p(k)$ of obtaining k from a Poisson distribution with mean λ, using the formula given in Appendix I.

Random: *int* uniformDiscreteMass(*int* a, *int* b)

This function computes the probability $p(k)$ of obtaining k from a discrete random distribution from a to b, using the formula given in Appendix I.

Appendix III - An Example Model

```
<sbml xmlns:html="http://www.w3.org/1999/xhtml"
          xmlns="http://www.sbml.org/sbml/level3" level="2" version="1">
  <model id="telomodel">

    <listOfCompartments>
      <compartment id="cell" name="cell" size="1" />
    </listOfCompartments>

    <listOfSpecies>
      <species id="G" compartment="cell" initialAmount="0" />
      <species id="telo1p" compartment="cell" initialAmount="6000" />
      <species id="source" compartment="cell" initialAmount="1"
                                              boundaryCondition="true" />
    </listOfSpecies>

    <listOfParameters>
      <parameter id="k0" value="1.0" />
    </listOfParameters>

    <listOfReactions>
      <reaction id="cellGrowth" name="cellGrowth" reversible="false">
      <listOfReactants>
        <speciesReference species="source" stoichiometry="0" />
      </listOfReactants>
      <listOfProducts>
        <speciesReference species="G" />
      </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <ci> k0 </ci>
          </math>
        </kineticLaw>
      </reaction>
    </listOfReactions>

    <listOfEvents>
      <event id="telomere_shorten">
        <trigger>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <and />
              <apply>
                <gt />
                <ci> G </ci>
                <cn type="integer"> 100 </cn>
              </apply>
```

```
          <apply>
            <gt />
            <ci> telo1p </ci>
            <cn type="integer"> 1000 </cn>
          </apply>
        </apply>
      </math>
    </trigger>
    <listOfEventAssignments>
      <eventAssignment variable="telo1p">
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <apply>
            <minus/>
            <ci>telo1p</ci>
            <apply>
              <csymbol encoding="text" definitionURL="http://www.sbml.org/
                                        sbml/symbols/uniformDiscreteRandom">
                uniformDiscreteRandom
              </csymbol>
              <cn>0</cn>
              <cn>1000</cn>
            </apply>
         </apply>
        </math>
      </eventAssignment>
      <eventAssignment variable="G">
        <math xmlns="http://www.w3.org/1998/Math/MathML">
          <cn type="integer"> 0 </cn>
        </math>
      </eventAssignment>
    </listOfEventAssignments>
  </event>
</listOfEvents>

 </model>
</sbml>
```