



RCCI Vol. 2, No. 1-2 ENERO- JUNIO, 2008 p. 21-27

Editor de ecuaciones para la plataforma de simulación de sistemas biológicos

Equation editor for the biological systems
simulation platform

**Gilberto Arias Naranjo¹, Noel Moreno Lemus¹ y Kalet León
Monzón²**

¹Facultad de Bioinformática, Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2½ Torrens, Boyeros, Ciudad de La Habana, Cuba.

²Centro de Inmunología Molecular, Calle 216, esquina 15, Atabey, Playa, Ciudad de La Habana, Cuba.
Autores para la correspondencia {gilberto, noel}@uci.cu; kalet@cim.sld.cu

Resumen

El surgimiento de la Biología de Sistemas introdujo el uso de herramientas computacionales en el estudio y modelación de sistemas biológicos. Una de estas herramientas es la plataforma de simulación de sistemas biológicos BioSyS, la cual realiza exploraciones intensivas en modelos de sistemas biológicos y permite realizar meta análisis sobre los resultados que estas exploraciones arrojan. En este trabajo se presenta un software que se integra a la plataforma BioSyS y permite la edición de sistemas de ecuaciones a través de una interfaz amigable que representa las expresiones mediante una notación simbólica. La herramienta cuenta además con una biblioteca de expresiones dando la posibilidad al usuario de conservar los fragmentos de fórmulas que más utiliza, exporta a diferentes formatos, incluido el estándar de representación matemática MathML y brinda mecanismos para analizar la homogeneidad dimensional de las ecuaciones escritas ayudando así a los investigadores en la revisión de las mismas.

Palabras clave: Análisis simbólico, editor ecuaciones, homogeneidad dimensional.

Abstract

The emergence of Systems Biology introduced the use of computational tools in the study and modeling of biological systems. One of these tools is the platform simulation of biological systems BioSyS, which conducts intensive explorations on models of biological systems and allows the performance of the meta-analysis of the results. This paper presents a software that is integrated into the platform BioSyS and allows editing equations systems through a user-friendly interface that represents expressions via a symbolic notation. The software also includes a library of expressions giving the user the ability to retain fragments of formulas commonly used; it exports to different formats, including the MathML standard for mathematical representation and provides mechanisms to analyze the dimensional homogeneity of the equations written, helping the researchers to review them.

Keywords: Dimensional homogeneity, equation editor, symbolic analysis.

Introducción

En los últimos años, la Biología ha experimentado una transición, de una era pobre de datos y de formas tradicionales de almacenarlos, a un mundo más rico en datos y donde la interpretación de estos es fundamental. Este cambio de paradigma ha dado nacimiento a un nuevo campo, la Biología de Sistemas. La unidad fundamental de la Biología de Sistemas es un modelo computacional que es construido a partir de componentes celulares y sus mecanismos de comunicación. La Biología de Sistemas brinda una oportunidad única de entrelazar moléculas y sus mensajes dentro de un aparato computacional que puede ser probado contra un amplio conjunto de condiciones y combinaciones.

En un escenario ideal la comunidad de investigadores necesita una herramienta computacional que integre y modele gráficamente información de modelos biológicos, efectúe simulaciones en gran escala y con una alta capacidad de cálculo, tenga una interfaz amigable (fundamentalmente para los no especialistas) y ofrezca servicios sobre la Web. Además, a medida que la complejidad de los modelos crezca, debe permitir que la plataforma de modelado pueda moverse desde un ambiente de cómputo en serie a uno distribuido (Jardon, 2007).

Con vistas a alcanzar este escenario ideal un equipo de investigadores del Centro de Inmunología Molecular (CIM), en conjunto con un grupo de desarrolladores de la Universidad de las Ciencias Informáticas (UCI), desarrollaron una investigación que arrojó como resultado la versión 1.0 de la plataforma de simulación de sistemas biológicos BioSyS (Moreno, 2007).

Los modelos en BioSyS son descritos mediante sistemas de ecuaciones diferenciales. Para potenciar el trabajo con la plataforma, se necesita que esta cuente con mecanismos que faciliten la edición de los sistemas de ecuaciones y además que permitan ayudar a los investigadores en el análisis de homogeneidad dimensional de los mismos.

Actualmente son muchas las herramientas que permiten la edición de ecuaciones. Dentro de las más populares encontramos a MathType (Chartwell-Yorke, 2008), WebEQ, MathML.NET Control, Kformula, OpenOffice Math. Todas ellas brindan interfaces amigables que facilitan la edición de ecuaciones al estilo "lo que ves es lo que ob-

tienes”, permiten exportar a gran variedad de formatos, sin embargo la mayoría no garantiza que las fórmulas escritas tengan una estructura sintáctica correcta, ni cuentan con herramientas de análisis simbólico, necesarias para realizar análisis de homogeneidad dimensional.

También existen herramientas que permiten realizar análisis simbólico de las ecuaciones descritas, como Mathematica y Maxima (Lutus, 2007), pero la edición de ecuaciones en ellas es engorrosa y no cuentan con mecanismos explícitos para realizar el análisis de homogeneidad dimensional.

En resumen ninguno de los software revisados integra todas las funcionalidades requeridas por BioSys lo que conlleva al desarrollo de una nueva aplicación computacional. En el siguiente trabajo se describe dicha herramienta.

Materiales y Métodos

Uno de los principales intereses fue contar con una herramienta multiplataforma. Es por ello que el lenguaje seleccionado para el desarrollo de la aplicación fue JAVA. JAVA es un lenguaje cuya portabilidad está probada, además de ser simple, robusto y permitir la creación de aplicaciones siguiendo el paradigma Orientado a Objetos con rapidez.

Otros lenguajes utilizados en la aplicación fueron Maxima, MATLAB y MathML. Todos con el propósito de exportar las ecuaciones a dichos lenguajes y ampliar, de esta forma, el campo de acción de la herramienta.

El IDE seleccionado fue Eclipse dado su carácter multiplataforma y que en principio fue desarrollado para trabajar con el lenguaje JAVA.

Maxima, además, es un asistente que utiliza álgebra computacional para el análisis de expresiones aritméticas. Los algoritmos de álgebra computacional pueden ser utilizados para integración o diferenciación simbólica, sustitución de una expresión por otra, simplificación. La aplicación a desarrollar debe contar con funcionalidades que le permitan comparar y simplificar expresiones para poder cumplir con el análisis de homogeneidad dimensional que se quiere. Es por ello que se decide utilizar Maxima como herramienta de soporte. Otra característica importante es que el código fuente de Maxima está disponible de forma gratuita, lo que posibilita que pueda ser compilado y utilizado en cualquier plataforma.

Una verdad invariable en el desarrollo de software es que no importa cuán bien se haya diseñado una aplicación, con el tiempo esta tiene que cambiar y crecer o está destinada al desuso. Mediante la utilización de patrones de diseño se puede lograr que los cambios que deben sufrir las aplicaciones para evolucionar sean fáciles de realizar y modificando la menor cantidad de código posible. En el desarrollo de la aplicación se aplicaron fundamentalmente dos patrones de diseño, Modelo Vista Controlador y el patrón Observador. Con la utilización de estos patrones se garantiza que la aplicación pueda extenderse con nuevas operaciones matemáticas con el menor esfuerzo posible.

Se utilizaron además estructuras de datos propias del proceso de compilación: tablas de símbolos y árboles de derivación. A las tablas de símbolos se hace referencia más adelante. Los árboles de derivación son generados de forma implícita a medida que se van editando las ecuaciones y son el mecanismo que simplifica el proceso de exportación a los diferentes lenguajes y garantizan que la inclusión de nuevos formatos de exportación sea una tarea sencilla y rápida de ejecutar.

Resultados y Discusión

Diseño de Clases

Se partirá mostrando el diagrama de clases que se utiliza en la implementación de la herramienta (ver Figura 1). Aclarar que el sistema cuenta con más de 70 clases, es por ello que se ha decidido limitar el diagrama sólo a aquellas que son necesarias para la comprensión y análisis del trabajo realizado.

Todo el trabajo con las entidades matemáticas lo realizan las clases que pertenecen a la jerarquía que tiene como clase base a BasicBox. Con ellas no sólo se representan las ecuaciones matemáticas, sino que intervienen en el proceso de conformación de las diferentes sentencias para los diferentes formatos a los que exporta la aplicación.

La clase EquationController es quien controla todo el proceso de interacción entre el modelo que se va generando y la interfaz visual donde se muestran las ecuaciones y son realizadas todas las operaciones. Además se encarga de interrelacionar todo el trabajo con variables, bibliotecas de expresiones y análisis de dimensiones con el modelo matemático que se está editando.

A continuación se realiza un análisis de las funcionalidades con que cuenta la aplicación.

Edición de Ecuaciones

Lo más importante en un proceso de edición de fórmulas matemáticas es que el usuario sienta que lo que escribe es lo mismo que se muestra. Con el editor de ecuaciones se logra una interfaz agradable, mostrando de forma simbólica todas las expresiones aritméticas que se introducen, además garantiza que las expresiones que se editen estén bien estructuradas sintácticamente.

El peso fundamental del editor de ecuaciones lo tienen las clases que pertenecen a la jerarquía cuya clase base es BasicBox. La estructura con que se diseñó se basa en la forma en que MathML representa sus operaciones. Para una mejor comprensión del texto que sigue serán tratados los operadores y las funciones (trigonométricas, logaritmos, etc.) con el término de operaciones. Con el mismo propósito las clases que se utilizan para representar las expresiones serán nombradas con el término de cajas, esto se debe a que gráficamente parece que se está trabajando dentro de cajas.

Para diseñar esta jerarquía, con la cual se representan todas las operaciones con las que se puede trabajar en el editor, se tuvieron en cuenta los siguientes requisitos:

- El operador menos (-) se considera como un operador unario y se aplica para cambiar el signo a términos en expresiones. Por tanto cuando se representa la operación "2 - 3", internamente se estaría generando algo como esto "2 + (- 3)".
- Para definir las expresiones se utiliza la definición

recursiva siguiente:

•Una expresión se representa como un término o una suma de términos.

•Un término se representa como un factor, un producto de factores o un término negado (antecedido por un signo negativo).

•Un factor es, una variable, una constante, una operación diferente a suma, producto y negación o una expresión encerrada entre paréntesis.

La estructura en la que se representan las expresiones es muy similar a una muy utilizada para representar gramáticas que generan expresiones aritméticas, sólo que esa variante incluye la resta en la regla de expresión, como operación entre términos y la división en la regla de término, como operación entre factores.

Se incluye la división en la declaración de un factor por problemas de representación visual, siendo posible mostrar, un numerador y denominador determinados, de diferentes formas (ver Figura 2). Con la variante tradicional de generación, no pueden ser diferenciadas las distintas representaciones visuales, sin embargo el diseño que se sigue en la aplicación sí contempla cualquier variante posible.

$$3 \frac{x}{2} \quad \frac{3x}{2} \quad \frac{3}{2}x$$

Fig. 2. Formas de representar gráficamente $3x/2$.

Las consideraciones tomadas en cuenta con el signo de menos obedecen a la necesidad de simplificar el trabajo de generación de las cadenas en

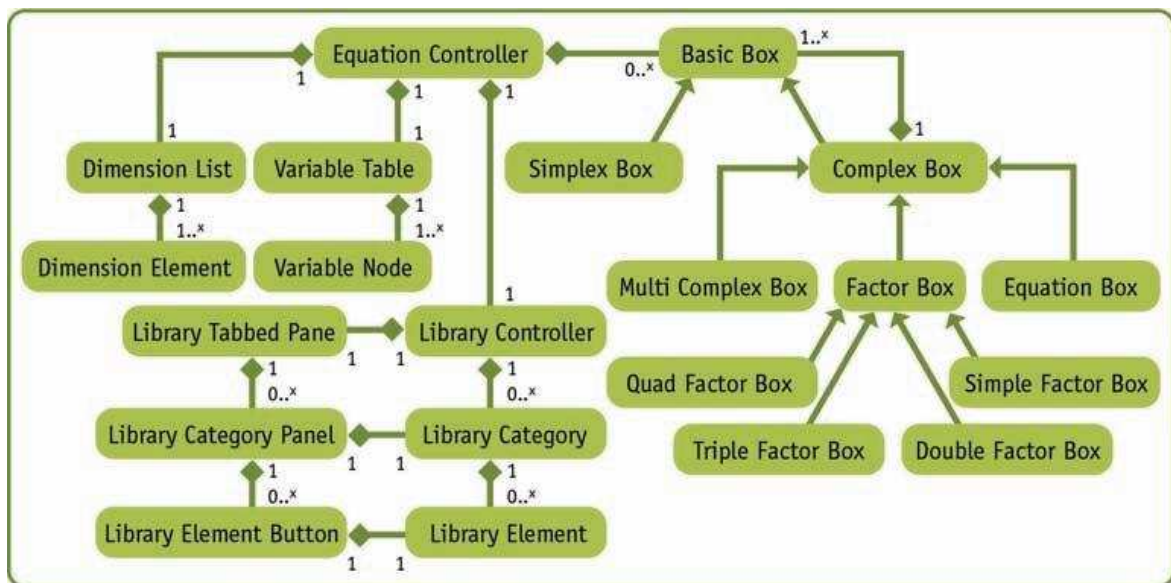


Fig. 1. Diagrama de clases. En la Figura aparece un subconjunto de las más de 70 clases con que cuenta el sistema.

formato MathML, MATLAB y Maxima. Aunque la jerarquía es extensa, existen clases que se utilizan como formas de abstracción de las clases que representan específicamente las operaciones y operandos; por tanto se considera pertinente explicar solamente las características y funcionalidades que estas definen, pues no es necesario entrar en particularidades para comprender el diseño implementado.

BasicBox: Es la clase base de la jerarquía de clases utilizadas por el editor de ecuaciones. Esta clase declara todas las funcionalidades que deben implementar las demás clases de la jerarquía, incluyendo las necesarias para poder realizar la exportación a los formatos MathML, MATLAB y para generar las instrucciones de Maxima que se utilizan en el análisis de dimensiones.

SimplexBox: Es la caja más simple de la jerarquía y se hará referencia a ella como caja simple. Con ella se representan los operandos de las expresiones que se editan. Cada vez que se genera una operación esta incluye todas las cajas simples necesarias para que sea una operación bien estructurada sintácticamente y no se considerará la fórmula como completa mientras existan cajas simples sin valor (vacías).

ComplexBox: Todas las cajas que se utilizan para representar operaciones heredan de ComplexBox, se hará referencia a ellas como cajas complejas. Esta clase tiene una relación de composición con la clase base BasicBox, esto significa que las cajas contenidas pueden ser simples u otras cajas complejas, así por ejemplo, para representar la expresión $\frac{x+y}{2}$ se tendría una caja división que contiene dos cajas, como numerador una caja compleja suma y en el denominador una caja simple con el valor "2", a su vez la caja compleja suma, del numerador, contiene dos cajas simples con los valores "y" y "x". Todas las cajas complejas definen la forma en que será representado visualmente su operador (o nombre, en el caso de las funciones) y además definen en qué posición estarán ubicadas las cajas contenidas en ellas.

MultiComplexBox: Como se dijo, se utilizó para la representación de las operaciones una concepción similar a la de MathML. MathML considera que los operadores de suma y multiplicación son n-arios, lo que significa que pueden tener una cantidad indeterminada de operandos. Esta caja compleja fue diseñada para agrupar todas las cajas que cumplen con esta característica, en este caso estarían ExpressionBox (para represen-

tar las expresiones) y TermBox (para representar los términos). Es importante aclarar que la caja TermBox incluye un atributo que representa el signo que tiene el término.

FactorBox: Con esta clase se engloban todas las cajas que representan operaciones diferentes a la suma, al producto y a la negación. De esta clase heredan SimpleFactorBox, DoubleFactorBox, TripleFactorBox y QuadFactorBox, las cuales se utilizan para representar las cajas con una, dos, tres y cuatro cajas componentes, respectivamente. Se hizo esta distinción de clases pues muchos comportamientos especificados en ComplexBox tienen una forma única de representarse para las cajas que tienen como base la misma clase, ya sea SimpleFactorBox, DoubleFactorBox o las restantes. Además se trató de lograr un diseño lo más generalizado posible previendo que nuevas funcionalidades en versiones superiores, requiriesen luego agregar niveles de abstracción en la jerarquía.

EquationBox: Si se está desarrollando un editor de ecuaciones es importante entonces que existan clases que representen las ecuaciones. Aunque con EquationBox se puede representar cualquier tipo de ecuación se han incluido dos cajas para expresar específicamente ecuaciones diferenciales y ecuaciones para definir funciones (ej. $f(x) = x$). Estas cajas tienen un peso importante pues todo lo que se describe en el editor son ecuaciones, por tanto cualquier expresión que se represente con las clases antes descritas estarán invariablemente contenidas dentro de una ecuación.

Exportando a otros formatos

Una de las tareas que potencia el uso de una herramienta es que esta sea capaz de exportar a formatos estándares en el campo de acción de la misma. Con tal propósito la herramienta permite exportar a formato MathML. La aplicación incluye además una opción para exportar las ecuaciones a formato MATLAB. Como la plataforma BioSys realiza la simulación de los sistemas apoyándose en el asistente MATLAB, la opción de exportar a dicho lenguaje garantiza un primer nivel de integración del editor a la plataforma.

En la jerarquía de clases que tiene como base la clase BasicBox, todas las clases incluyen métodos que devuelven la fórmula en un formato determinado (ej. getFormulaMaxima, getFormulaContentMathML, etc.). En el caso de la caja simple (SimplexBox) estos métodos devuelven el valor que esta contiene en el formato necesario. En el caso de una caja compleja (cualquiera que

herede de ComplexBox), el método construye la solución a partir del resultado que devuelve el mismo método en las cajas contenidas dentro de ella, si alguna de estas es compleja el proceso se repite hasta llegar a cajas simples.

Biblioteca de expresiones

La biblioteca de expresiones permite que los investigadores puedan acceder de manera rápida a fórmulas que utilizan de forma frecuente, lo que convierte este elemento en una potencialidad de la herramienta. Además brinda la posibilidad de agrupar las fórmulas por categorías previamente definidas por el usuario, lo que agiliza la búsqueda de las mismas.

Análisis de homogeneidad dimensional

El objetivo de incluir funcionalidades para el análisis de homogeneidad dimensional fue hacer transparente, para los investigadores, el uso de herramientas de análisis simbólico.

El primer elemento a tener en consideración es que el usuario tiene que definir cuáles son las dimensiones que tendrá cada una de las variables del sistema de ecuaciones que está editando. Las variables son extraídas de forma automática por la aplicación a partir de las cajas simples que las contienen y se almacenan en una tabla de símbolos. Esta tabla contiene además el estado de las variables (si tienen dimensión asignada o no) y en caso de tener una dimensión asignada cuál es.

El proceso de comprobación de homogeneidad dimensional se realiza utilizando el asistente matemático Maxima. Las funciones de Maxima que se utilizan para realizar el mismo son las siguientes:

- `writefile (filename)` para abrir el fichero en que serán almacenados los resultados. Es la primera instrucción en ejecutarse.
- `batchload (filename)` para ejecutar el fichero con las instrucciones. Se ejecuta luego de abrir el fichero en el que serán salvados los resultados.
- `closefile (filename)`: al finalizar `batchload` se cierra el fichero con los resultados utilizando este comando.
- `print (expr)`: muestra el resultado de la expresión `expr`. Es importante utilizar esta función pues `batchload` no muestra salidas salvo las mostradas por `print` y `describe`.
- `string (expr)`: devuelve la expresión `expr` en su forma lineal.
- `fullratsimp (expr)`: simplifica la expresión `expr`

y continúa repitiendo el proceso de simplificación con el resultado obtenido hasta que este no sufra modificaciones de una simplificación a otra.

Para explicar el proceso que se sigue será utilizada la siguiente ecuación:

$$\frac{dn}{dt} = ks + \frac{kp \cdot A \cdot n}{A + n} - kd \cdot n$$

La cual se utiliza, en problemas de inmunología, para representar el crecimiento de una población de células n en el tiempo, con una fuente externa ks , una tasa de muertes kd y una proliferación dependiente de la interacción con otra población A , a una tasa máxima kp . En el ejemplo las variables toman las siguientes dimensiones:

- n y A son variables de masa, se hará una generalización asignándoles la letra "m",
- t expresa tiempo y le será asignada la dimensión "t",
- kp y kd se expresan en la inversa del tiempo, por tanto serían "t⁻¹" y
- ks se expresa en masa contra tiempo, quedando "m/t".

Será asumido que ya la fórmula fue introducida, así como, las dimensiones y las asociaciones entre variables y dimensiones.

El primer paso es generar el fichero que contiene las instrucciones que serán ejecutadas por el comando `batchload`. Se utilizará para comprobar la equivalencia de las expresiones el comando `fullratsimp`.

Se descompone el miembro derecho en términos. Para ello se cuenta con dos casos:

1. Si el miembro derecho es una caja simple, un factor (cualquier caja que herede de FactorBox) o un término (TermBox). Entonces el miembro derecho está compuesto por un solo término.
2. Si el miembro derecho es una expresión (ExpressionBox) entonces los términos que lo componen están dados por las cajas que están contenidas dentro de la caja expresión (ver Figura 3).

$$\frac{dn}{dt} = ks + \frac{kp \cdot A \cdot n}{A + n} - kd \cdot n$$

Fig. 3. Descomposición de una expresión en términos. Se encierran en un cuadro los diferentes términos en que es descompuesto el miembro derecho de la ecuación.

Se asume que la dimensión del miembro izquierdo de la ecuación es la deseada. En ese caso se toman los términos del miembro derecho y se verifica que cada uno tenga una dimensión equivalente a la del miembro izquierdo. Para ello se sustituyen las variables por las dimensiones en cada uno de los términos y se construyen instrucciones en Maxima con la siguiente estructura:

```
string (fullratsimp(miembroizquierdo / terminodelmiembroderecho) )
```

Como el comando batchload no muestra las salidas, salvo las que generan print y describe entonces se encierran las instrucciones dentro de comandos print, el resultado final para el ejemplo sería:

```
print (string (fullratsimp((m/t) / (m/t))))$
print (string (fullratsimp ((m/t) / ((t^(-1))*m*m)/(m+m))))$
print (string (fullratsimp ((m/t) / (t^(-1)*m))))$
```

Estas instrucciones se almacenan en un fichero que será llamado "formulas.max".

El segundo paso es generar el fichero que será pasado al Maxima para realizar los cálculos y obtener los resultados. Este fichero será nombrado "calculus.max" y contendrá las siguientes instrucciones:

```
writefile("results.max")$
batchload("formulas.max")$
closefile();
quit();
```

Al ejecutarse el Maxima pasando como parámetro el archivo "calculus.max" se obtendrá otro archivo de nombre "results.mx" el cual contendrá varias líneas (una por cada instrucción ejecutada en "formulas.max") con números o expresiones más complejas.

Si el valor obtenido es un número entero o en forma de fracción (ej. $3/2$), significa que la expresión del término del miembro derecho es igual a la expresión del miembro izquierdo multiplicada por una constante, lo que implica que la dimensión del mismo es correcta. En caso contrario la dimensión del término del miembro derecho no es la correcta. Para el ejemplo dado el contenido del archivo "results.max" serán tres números, por tanto la ecuación está bien formulada en términos de sus dimensiones.

El procedimiento antes descrito se ejecuta para cada una de las ecuaciones que componen el sistema de ecuaciones que se está editando. La herramienta muestra en color rojo, las expresiones que se considera el investigador debe revisar por posibles errores en las dimensiones.

Conclusiones

La principal potencialidad de la herramienta es que cuenta con facilidades visuales y funcionalidades que permiten al investigador centrarse en el problema que están resolviendo y no en las dificultades que pueda presentar en la representación de los sistemas de ecuaciones que utiliza y en revisiones de la estructura sintáctica de los mismos. El hecho de contar con funcionalidades para el análisis de la homogeneidad dimensional de las fórmulas hace transparente el trabajo con asistentes matemáticos haciendo que las personas que utilicen la herramienta no requieran grandes conocimientos sobre los mismos.

Aunque la herramienta fue concebida para editar sistemas de ecuaciones que modelen sistemas biológicos, las operaciones matemáticas son genéricas. Este hecho, unido a la posibilidad de exportar a varios formatos, permite que el editor pueda ser utilizado por especialistas de otras ramas. Además, puede ser extendido de forma simple con nuevas operaciones y facilidades de exportación, lo que constituye una fortaleza.

Referencias

- Chartwell-Yorke Mathematics ICT. MathType 6.5. 2008, Disponible en: [<http://www.chartwellyorke.com/mathtype.html>]
- Jardon, M. System Biology: An Overview. 2007, Disponible en: [<http://www.scq.ubc.ca/systems-biology-an-overview/>].
- Lutus, P. Symbolic Mathematics Using Maxima. 2007, Disponible en: [<http://arachnoid.com/maxima/index.html>].
- Moreno Lemus, N. (et. al.). BioSyS: Software para la simulación y análisis de sistemas biológicos. 2007 Disponible en: [<http://biblioteca.uci.cu/uciencia/tbio.html>].