Conservation analysis of large biochemical networks

Ravishankar Rao Vallabhajosyula*, Vijay Chickarmane and Herbert M. Sauro
Keck Graduate Institute, 535 Watson Drive, Claremont, CA 91711, USA

Received on September 23, 2005; revised on November 22, 2005; accepted on November 23, 2005
Advance Access publication November 29, 2005
Associate Editor: Martin Bishop

ABSTRACT

Motivation: Large biochemical networks pose a unique challenge from the point of view of evaluating conservation laws. The computational problem in most cases exceeds the capability of available software tools, often resulting in inaccurate computation of the number and form of conserved cycles. Such errors have profound effects on subsequent calculations, particularly in the evaluation of the Jacobian which is a critical quantity in many other calculations. The goal of this paper is to outline a new algorithm that is computationally efficient and robust at extracting the correct conservation laws for very large biochemical networks.

Results: We show that our algorithm can perform the conservation analysis of large biochemical networks, and can evaluate the correct conserved cycles when compared with other similar software tools. Biochemical simulators such as Jarnac and COPASI are successful at extracting only a subset of the conservation laws that our algorithm can. This is illustrated with examples for some large networks which show the advantages of our method.

Availability: The software is available as part of the latest release of Systems Biology Workbench (SBW version 2.5.0) and can be downloaded from http://www.sys-bio.org. The software is licensed under the BSD open source license and is freely available at sourceforge.

Contact: rrao@kgi.edu

1 INTRODUCTION

One of the characteristics of biological networks is the conservation of certain molecular subgroups, termed moieties (Reich and Selkov, 1981). A typical example of a conserved group is the conservation of the adenine nucleotide moiety, i.e. the total amount of ATP, ADP and AMP is constant during the evolution of the system. Other more common examples include the conservation of protein between the phosphorylated and unphosphorylated states. Over long time periods the conservations do not hold as other slower processes can affect the way we carry out certain numerical procedures. The Jacobian matrix in particular is a central quantity that is required in the computation of many analyzes. Examples include, Bifurcation Analysis (Chickarmane et al., 2005), Frequency Analysis, Metabolic Control Analysis (Reder, 1988) and more traditional approaches such as the numerical integration of stiff differential equations, steady-state analysis and certain optimization approaches. All these analyzes depend on the computation of a non-singular Jacobian matrix. When a model contains conserved moieties, the Jacobian becomes singular and such analyzes are no longer possible. Evaluating the conservation laws is therefore critical. Finally, taking into account conservation laws also permits a simulator to reduce the size of a model and thereby increase the performance of the software.

The determination of conservation laws in biochemical network models has been a subject of discussion for many years. The interest in this topic can be traced back to the study of stoichiometric networks by the chemical engineering community in the 1960s (for a review, see Sauro and Ingalls, 2004). Studies of relevance include works by Aris (1965) to be followed by Horn and Jackson (1972), Clarke (1980) and Feinberg (1989). Many of the early simulators, such as SCAMP (Sauro, 1993) and Gepasi (Mendes, 1993) specifically incorporated algorithms to compute the conservation laws both as a means to reduce the model size and more importantly as a means to compute the reduced Jacobian. None of the approaches taken by these early simulators were robust for large networks, indeed as we will show, a number of contemporary simulators suffer the same problem.1

Traditionally, most simulators have implemented Gaussian Elimination (Hofmeyr, 1986; Holstein and Greenshaw, 1994; Heinrich and Schuster, 1996; Cornish-Bowden and Hofmeyr, 2002; Sauro and Ingalls, 2004) to extract the conservation laws. Gaussian Elimination is an easy and fast algorithm to implement and is well suited for small networks. However, it is not applicable to large networks as errors tend to accumulate in the computation of pivots and the subsequent elimination of rows. An alternative to avoid round-off errors involves the use of integer arithmetic. This approach, while being accurate can be computationally slow for large systems.

The development of a robust method of computing the conservation laws for biochemical networks is important for large systems. It is in this regard that the US Department of Energy sponsored GTL (see http://doegenomestolife.org/) program has invested in the development of robust means of handling large biochemical systems. This is a very critical step for the GTL program that envisages among its goals the development of advanced computational methods and capabilities to further the understanding of complex biological systems and predict their behavior. This paper therefore addresses the problem of reliable computation of conservation laws for large networks by presenting a new method based on Householder QR factorization (Householder, 1958). We illustrate

1It should be noted that the majority of simulators for systems biology do not attempt a conservation analysis.
this method with a simple example and compare its ability to extract the correct conservation laws for large systems with other similar tools. Various applications of this method are also addressed.

2 CONSERVATION ANALYSIS

In the following, we adopt the nomenclature pertaining to biochemical networks established by Reder (1988). A biochemical network involves reactions between various species whose time evolution can be described by

\[
\frac{d\mathbf{S}(t)}{dt} = \mathbf{Nv}(t),
\]

(1)

where \( \mathbf{S}(t) = [S_1(t) S_2(t) \cdots S_m(t)]^T \) is the vector of time-dependent species concentrations, \( \mathbf{N} \) is the Stoichiometric matrix relating the species to the reactions they participate in and \( \mathbf{v}(t) = [v_1(t) v_2(t) \cdots v_n(t)]^T \) is the vector of rates of the reactions that comprise the network. Consequently \( \mathbf{N} \) is a matrix with \( m \) rows and \( n \) columns, where each column corresponds to a particular reaction and each row corresponds to a particular species. Equation (1) therefore describes the dynamics of the network species in relation to the reactions. In many networks however, some of the reactions are formed such that certain conservation relations naturally follow. These relations can be interpreted as dependencies among species or alternatively, as dependencies in the rows of the \( \mathbf{N} \) matrix. This means the rank of the \( \mathbf{N} \) matrix is less than what it would be without the conservation relations. Let us denote this rank by \( m_0 \). The goal of any algorithm that identifies these conservation relations should be to identify and separate independent and dependent rows of the stoichiometry matrix, or equivalently partition the species participating in the biochemical network into dependent and independent entities. It follows that there are \( m_0 \) independent species and \( m - m_0 \) dependent species. Let us denote these as \( \mathbf{S}_d(t) \) and \( \mathbf{S}_0(t) \). The time evolution of the biochemical network described in Equation (1) can therefore be rewritten as

\[
\frac{d\mathbf{S}(t)}{dt} = \frac{d\mathbf{S}_d(t)}{dt} \quad \text{and} \quad \frac{d\mathbf{S}_0(t)}{dt} = \mathbf{N}_0 \mathbf{v}(t),
\]

(2)

where the \( m \times m_0 \) matrix \( \mathbf{L} \) is called the Link matrix (Reder, 1988). The dynamics of the full system described by Equation (1) can therefore be partitioned into two components, one describing the dynamics of the independent species and the other corresponding to the dependent species that derives its dynamics from the independent species. These can be written as

\[
\frac{d\mathbf{S}_d(t)}{dt} = \mathbf{N}_R \mathbf{v}(t) \quad \text{and} \quad \frac{d\mathbf{S}_0(t)}{dt} = \mathbf{L}_0 \mathbf{N}_R \mathbf{v}(t).
\]

(3)

Further simplification of Equation (4) yields the relation between time evolution of dependent and independent species as

\[
\frac{d\mathbf{S}_d(t)}{dt} - \mathbf{L}_0 \frac{d\mathbf{S}_d(t)}{dt} = 0.
\]

(5)

It follows upon integrating Equation (5) that \( \mathbf{S}_d \) and \( \mathbf{S}_0 \) are related by a constant vector \( \mathbf{T} = [T_1 T_2 \cdots T_m] \) such that

\[
\mathbf{S}_d(t) - \mathbf{L}_0 \mathbf{S}_d(t) = \left[ -\mathbf{L}_0 \mathbf{I} \right] \left[ \begin{array}{c} \mathbf{S}_d(t) \\ \mathbf{S}_0(t) \end{array} \right] = \mathbf{T}.
\]

(6)

Introducing a matrix \( \Gamma \) in place of \( [ -\mathbf{L}_0 \mathbf{I}] \) in Equation (6), we can deduce that \( \mathbf{TS} = \mathbf{T} \). The \( (m - m_0) \times n \) matrix \( \Gamma \) is called the conservation matrix, as it relates the species vector \( \mathbf{S} \) to the vector of conserved moieties \( \mathbf{T} \). These represent molecular subgroups that are conserved during the evolution of the network (Reich and Selkov, 1981). Each row of the conservation matrix \( \Gamma \) represents a distinct conserved cycle of the network. The values of the vector \( \mathbf{T} \) can in practice be obtained by substituting the initial conditions of the species into the relation \( \mathbf{T} = \mathbf{S}_d(0) - \mathbf{L}_0 \mathbf{S}_0(0) \).

Another aspect of interest in the study of biochemical networks relates to the evaluation of the Jacobian Matrix. The dynamics of the network described by Equation (1) are most often non-linear, and hence a linearized system can be obtained by studying the perturbation around a steady-state value. Ignoring second-order terms, a linear equivalent of Equation (1) can be obtained as

\[
\frac{d(\mathbf{S})}{dt} = \mathbf{J}_F \delta \mathbf{S},
\]

(7)

where \( \mathbf{J}_F \) is the Jacobian matrix for the full system and \( \delta \mathbf{S} \) represents infinitesimal changes in \( \mathbf{S} \). The dependencies in the network between species are reflected in the Jacobian matrix \( \mathbf{J}_F \) being singular. The full Jacobian matrix can therefore be built using the relationship between dependent and independent species, given by \( \mathbf{L} \) and \( \mathbf{N}_R \) matrices. This can be shown after modifying an expression given by Heinrich and Schuster (1996, p. 40) to be

\[
\mathbf{J}_F = \mathbf{N} \mathbf{\epsilon} = \mathbf{L} \mathbf{N}_R \mathbf{\epsilon},
\]

(8)

where \( \mathbf{\epsilon} \) is the Elasticity coefficient matrix, with \( \epsilon_{ij} = \partial v_i / \partial S_j \). A non-singular reduced Jacobian matrix, \( \mathbf{J}_R \), can be constructed using conservation analysis. Indeed, it can be shown, using Equations (2) and (7), that \( \mathbf{J}_R \) is given by the relation \( \mathbf{J}_R = \mathbf{N}_R \mathbf{\epsilon} \mathbf{L} \).

3 DEALING WITH LARGE SYSTEMS

The efficacy of a number of methods with varying degrees of computational complexity for conservation analysis have been reviewed by Sauro and Ingalls (2004). These include computing the Null space of \( \mathbf{N} \), Gauss–Jordan method, reduction to row echelon form and singular value decomposition (SVD). However, these methods are prone to errors when handling large networks. 

Sauro and Ingalls (2004) propose the use of SVD for such systems. However, even the SVD is inaccurate for large systems such as those used as examples later in this paper (see Fig. 1 for a metabolic network of \( \textit{E. coli} \)). The reason for this is that the condition number of the system, defined as the ratio of the largest to the smallest singular value, is very large for nearly singular matrices. That implies that the \( \mathbf{N}_R \) matrix would not be of full rank, as one or more dependent rows has not been eliminated. This results in one of
the solution vectors being closely aligned in the direction of the independent vectors, leading to errors when solutions are computed.

The objective of our paper is therefore to build a robust means of performing conservation analysis for large biochemical networks. We propose that a numerical tool with such capability can be built using the Householder QR method (Householder, 1958; Hansen, 1992). The QR method factorizes a given matrix $A$ into an orthogonal matrix $Q$ and an upper triangular matrix $R$. The potential of this method to retain high numerical stability makes it an ideal candidate for conservation analysis. The applicability of the Householder QR method to biochemical networks has in fact been suggested earlier (Holstein and Greenshaw, 1994) but was never explored or developed. While the QR factorization can be obtained using a number of algorithms, including Givens Rotations, Gram–Schmidt method as well as the Householder method, the Householder method is more numerically stable than the other approaches since it uses orthogonal similarity transforms (Householder and Bauer, 1959). For a general $m \times n$ matrix, Gaussian elimination requires about $2n^3/3$ floating point operations (FLOPS). The Householder QR method on the other hand requires about $2n^2(m - n/3)$ FLOPS. This is also better than the QR factorization based on Givens rotations which requires nearly twice as many FLOPS due to additional computation of square roots. In practice, the Givens rotations method is useful only when a relatively few off diagonal elements need to be zeroed, as opposed to the whole lower triangular part of the matrix in Householder method.

4 QR DECOMPOSITION

A general non-symmetric $m \times n$ matrix $A$ can be written as the product of two matrices $Q$ and $R$ as

$$AP = QR,$$

where $Q$ is an $m \times m$ orthogonal matrix such that $Q^TQ = I$, the Identity matrix, $P$ is an $n \times n$ permutation matrix comprising column exchanges in $A$ and $R$ is an $m \times n$ upper trapezoidal matrix whose elements below the main diagonal are all zero. The permutation matrix $P$ arises from the column exchanges when the Householder method is employed to build the $Q$ matrix through successive reflections. This is a modification by Golub (1965) to the original triangularization algorithm of Householder (1958) and adds stability to the numerical method. It should be noted that the sum of squares of elements in $R$ matrix is the same as the sum of squares of elements in corresponding column, after taking the permutation matrix $P$ into account. The QR decomposition provides a highly stable algorithm that has been applied to the solution of least-squares problems (Cox and Higham, 1997) as well as the accurate computation of SVD (Higham, 2000). The QR decomposition is also used to build the QR algorithm which is used for robust Eigenvalue computation, see Golub and Van Loan (1996).

The orthogonal matrix $Q$ and the upper trapezoidal matrix $R$ contain information regarding the dependencies in the rows of $A$. This is of interest to the factorization of the stoichiometry matrix that defines a biochemical network. The key to conservation analysis is the identification of dependent and independent species. Since the rows in the stoichiometry matrix correspond to species, it can be noted that changing the order of the species is equivalent to changing the order of the rows. This indicates that the rows can be permuted until those corresponding to the independent species occupy the first $m_0$ rows yielding $N_R$, while those corresponding to the dependent species will comprise the bottom part of the matrix, yielding $N_D$. It should be noted that while a permutation of the rows of $A$ results in a similarly row-permuted $Q$, the sign of the values of $R$ in the permuted row is changed, which can be traced back to the way the QR algorithm is implemented. This indicates that if the $Q$ and $R$ matrices are partitioned into $m_0$ and $(m - m_0)$ blocks such that

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} U \\ 0 \end{bmatrix}$$

their product can be compared with the partitioning of the stoichiometric matrix $N$ into $N_R$ and $N_D$ leading to the relation

$$N = \begin{bmatrix} N_R \\ N_D \end{bmatrix} = \begin{bmatrix} Q_{11}U & U^T \\ Q_{21} & U^T \end{bmatrix}.$$ 

(11)

From the relationship $N_D = L_D N_R$ and Equation (11), it follows that $L_0$, $Q_{11}$ and $Q_{21}$ are related by $L_0 = Q_{21}Q_{11}^{-1}$. The matrix $L_0$ will exist if $Q_{11}$ is non-singular and hence invertible. A similar observation was made for the case of Gaussian Elimination by Holstein and Greenshaw (1994), who also briefly discussed the merits of the Householder QR factorization for large systems. In the case of the Householder QR however, we found that obtaining an invertible $Q_{11}$ starting with a general stoichiometric matrix is a slow process. This is due to the fact that after each row permutation, the singular values of $Q_{11}$ have to be obtained until there are no more zero singular values. The computational cost of repetitive calls to a SVD algorithm renders this approach impractical. In the following, we outline a new approach based on Householder QR for conservation analysis that is robust and that does not face the problem of repetitive singular value computation.

5 HOUSEHOLDER QR FACTORIZATION

The Householder method was first described by Householder (1958) as an algorithm to triangularize a general non-symmetric matrix. A geometric interpretation of this method entails that we see the network reactions as vectors in the space of species involved. The Householder method is a series of projections of the reactions onto a coordinate system constructed by sequential Householder reflections. These reflections have the effect of creating an upper trapezoidal matrix that can then be further reduced to obtain the

A Householder reflection can be defined as a transformation that takes any vector and reflects it about a plane. This plane can be constructed in a manner such that the reflected vector has certain desired properties. This has a simple geometrical interpretation—the method maps the original matrix, each column of which describes a vector— to another matrix where the elements below the diagonal in the column of interest have been made zero. This is done by replacing the column with a vector that reflects it in a particular plane.

The matrix, which in our case, is the stoichiometric matrix for the biochemical network consists of \( n \) columns of length \( m \) each. These can be treated as \( n \) vectors in an \( m \)-dimensional space. The Householder algorithm begins by reflecting the first vector onto the first axis of the \( m \)-dimensional coordinate system. This has the effect of setting all the elements of the first column of the stoichiometric matrix below the first element to zero. The reflection plane is perpendicular to a given vector \( v \), which is used to construct the Householder matrix \( H \). This matrix is given by \( H = I - 2vv^T \), where \( v \) is a normalized vector (\( |v|^2 = v^Tv = 1 \)). The Householder matrices are symmetric (\( H^T = H \)) and orthogonal (\( HH^T = I \)). Therefore \( H \) is a non-singular matrix which is its own inverse (\( H^{-1} = H^T \)). Each householder matrix eliminates the zeros below the diagonal for each column of the original matrix. Let us denote the first Householder matrix as \( H_1 \). This matrix can eliminate the elements below the diagonal in the first column of the matrix.

It can therefore be shown that for an \( m \times n \) stoichiometric matrix, \( n - 1 \) such Householder matrices are required to create an \( R \) matrix with zeros below the main diagonal. It follows that the product of the Householder matrices is the orthogonal matrix \( Q \) in Equation (9) is given as \( Q = H_1 H_2 \cdots H_{n-1} H_{n-1} \), where \( H_i \) is the Householder matrix that can annihilate elements below the diagonal in the \( i \)-th column of the stoichiometric matrix.

6 CONSERVATION MATRIX EVALUATION

The QR decomposition using Householder reflections is available as a part of the LAPACK library (Anderson et al., 1995) which is in the public domain. We used the CLAPACK version to build a module that would connect to the Systems Biology Workbench (SBW, Sauro et al., 2003). This allowed us to develop the conservation analysis module that first builds the Stoichiometric matrix \( N \) in full form for a chosen model. The key to obtaining the conservation relations in the model lies in using \( N^T \) to carry out a QR decomposition. This is because the permutation matrix \( P \) contains the new order of the columns of \( N^T \) that is needed to generate \( Q \) and \( R \) in a numerically stable manner. The new order of the columns in the transpose is equivalent to the new order of the rows in the original stoichiometric matrix. This can be expressed as \( N^T P = QR \), where the subscript \( i \) indicates that the matrices \( Q_i, R_i \) and \( P_i \) correspond to those obtained from the QR decomposition of the transpose of the original stoichiometric matrix. Since \( Q_i \) is an orthogonal matrix, multiplying both sides by \( Q_i^T \) leads to the following relation:

\[
Q_i^T N^T P_i = R_i. \tag{12}
\]

The above form has a matrix \( R_i \) on the right-hand side that is upper trapezoidal. Furthermore, if the stoichiometry matrix has rows that are not linearly independent, then \( R_i \) can be partitioned into four submatrices that are central to obtaining the conservation laws. The form of Equation (12) can be made simpler by scaling each row of \( R_i \) to have a unity on the main diagonal and then performing a Gauss–Jordan elimination to zero out the above diagonal elements. This can be restricted only to the non-zero rows of \( R_i \), yielding

\[
\begin{bmatrix}
I & M \\
0 & 0
\end{bmatrix}
\]

\[
\tag{13}
\]

This matrix has the row echelon form that has previously been described by Sauro and Ingalls (2004). It therefore follows that the \( L_0 \) matrix is given simply as \( L_0 = M^T \). The conservation matrix \( \Gamma \) can then be built from the relation \( \Gamma = [-L_0 I] \). The Link matrix can similarly be obtained using the relation in Equation (3).

The order of the species that satisfies the conservation relation \( \Gamma S = 0 \) can be obtained using the \( m \times m \) column permutation matrix \( P_c \). As observed earlier, \( P_c \) arises due to rearrangement of the columns of \( N^T \), or equivalently rearrangement of the rows of \( N \). Hence the new order of the species can be built by noting the indices of rows of \( N \) that have been interchanged. The matrices \( N_0 \) and \( N_i \) can then be obtained from the reordered stoichiometry matrix.

The QR decomposition for the stoichiometric matrix results in a very robust numerical scheme to extract the conserved cycles in the biochemical network. This can be seen from the results for some large network models in Table 1. These models were obtained from the website for in silico organisms at the Systems Biology Research Group at University of San Diego, California (see http://gcrg.ucsd.edu/organisms/). These models were built from genomic studies and are available as Microsoft Excel spreadsheets that we have converted into SBML models in this analysis were constructed with data from in silico Organisms website (http://gcrg.ucsd.edu/organisms/).

The table below shows the number of conserved cycles for some large network models obtained using different methods.

<table>
<thead>
<tr>
<th>Method name</th>
<th>iCS291a</th>
<th>iJE660b</th>
<th>jJR904c</th>
<th>jND750d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Householder QR</td>
<td>36</td>
<td>19</td>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>LU (with full pivoting)</td>
<td>36</td>
<td>19</td>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>LU (with partial pivoting)</td>
<td>36</td>
<td>19</td>
<td>42\textsuperscript{1}</td>
<td>100\textsuperscript{10}</td>
</tr>
<tr>
<td>PySCeS (Olivier et al., 2005)</td>
<td>36</td>
<td>19</td>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>Jarnac (Sauro, 2000)</td>
<td>36</td>
<td>19</td>
<td>41</td>
<td>98</td>
</tr>
<tr>
<td>COPASI</td>
<td>31</td>
<td>16</td>
<td>36</td>
<td>89</td>
</tr>
</tbody>
</table>

\textsuperscript{1}Helicobacter pylori, 396 Species, 381 Reactions (Schilling et al., 2002).
\textsuperscript{2}E.coli, 537 Species, 739 Reactions (http://gcrg.ucsd.edu/organisms/).
\textsuperscript{3}E.coli, 764 Species, 931 Reactions (Reed et al., 2003).
\textsuperscript{4}Saccharomyces cerevisiae, 1072 Species, 1149 Reactions (Dusarte et al., 2004).

This table shows that the QR factorization routine, and the permutation matrix returned by the routine can be used to reorder the original species into independent and dependent species lists.

Another point of interest is that the \( R \) matrix in the QR decomposition of a general \( m \times n \) matrix \( A \) can be related to \( A \) by

\[
Q_i^T N^T P_i = R_i. \tag{12}
\]
constructing the matrix product $A^T A$, which is a square matrix. We can then proceed to factorize this square matrix using Cholesky decomposition into the product of an upper triangular matrix and its transpose. It can then be shown that this upper triangular matrix is in fact $R$. However this approach is not preferred as dependencies in the rows of $A$ are reflected in the ill-conditionality of the product $A^T A$, making it difficult to construct $R$. It is hence desirable to compute $R$ using QR decomposition instead of using $A^T A$.

It is noted in brief that the procedure described in this section to obtain $L_0$ can also be used to obtain the $K_0$ matrix and the null Space (Olivier, 2005; Hofmeyr, 2000). However, instead of $N^T$ we should use $N$ to compute these matrices. The $K$ matrix is representative of the number of independent fluxes (Reder, 1988; Heinrich and Schuster, 1996; Klamt and Stelling, 2003).

7 AN EXAMPLE NETWORK

The Householder QR method described in this paper is now illustrated by means of an example to compute the QR factorization of the network as shown in Figure 2. The original order of species for this network is $[ES, E, S1, S2]$. The reactions $[v_1, v_2, v_3]$ are

\[ v_1 : ES \rightarrow E + S1; \quad v_2 : S1 \rightarrow S2; \quad v_3 : E + S2 \rightarrow ES. \]

The stoichiometric matrix $N$ and its transpose $N^T$ for this network can be constructed as

\[
N = \begin{bmatrix}
-1 & 0 & 1 \\
1 & 0 & -1 \\
1 & -1 & 0 \\
0 & 1 & -1
\end{bmatrix}, \quad N^T = \begin{bmatrix}
-1 & 1 & 1 & 0 \\
0 & 0 & -1 & 1 \\
1 & -1 & 0 & -1
\end{bmatrix}.
\]

We now show how the conservation relations for this network can be extracted using the Householder QR approach. The factorization for $N^T$ is given by $N^T P_R = Q R_R$, where $P_R$ and $Q_R$ are

\[
P_R = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad Q_R = \begin{bmatrix}
-0.7071 & 0.4082 & 0.5774 \\
0 & -0.8165 & 0.5774 \\
0.7071 & 0.4082 & 0.5774
\end{bmatrix}.
\]

\[
R_R = \begin{bmatrix}
1.4142 & -0.7071 & -1.4142 & -0.7071 \\
0.0 & 1.2247 & 0.0 & -1.2247 \\
0.0 & 0.0 & 0.0 & 0.0
\end{bmatrix}.
\]

We now show how the conservation relations for this network can be extracted using the Householder QR approach. The factorization for $N^T$ is given by $N^T P_R = Q R_R$, where $P_R$ and $Q_R$ are

\[
P_R = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad Q_R = \begin{bmatrix}
-0.7071 & 0.4082 & 0.5774 \\
0 & -0.8165 & 0.5774 \\
0.7071 & 0.4082 & 0.5774
\end{bmatrix}.
\]

\[
R_R = \begin{bmatrix}
1.4142 & -0.7071 & -1.4142 & -0.7071 \\
0.0 & 1.2247 & 0.0 & -1.2247 \\
0.0 & 0.0 & 0.0 & 0.0
\end{bmatrix}.
\]

We now show how the conservation relations for this network can be extracted using the Householder QR approach. The factorization for $N^T$ is given by $N^T P_R = Q R_R$, where $P_R$ and $Q_R$ are

\[
P_R = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad Q_R = \begin{bmatrix}
-0.7071 & 0.4082 & 0.5774 \\
0 & -0.8165 & 0.5774 \\
0.7071 & 0.4082 & 0.5774
\end{bmatrix}.
\]

\[
R_R = \begin{bmatrix}
1.4142 & -0.7071 & -1.4142 & -0.7071 \\
0.0 & 1.2247 & 0.0 & -1.2247 \\
0.0 & 0.0 & 0.0 & 0.0
\end{bmatrix}.
\]

It should be kept in mind that $Q_R$ is built out of a product of Householder matrices, each of which results in zeroing elements below the main diagonal of $R_R$. It can be noted that the rank of the stoichiometric matrix $m_0$ is the number of rows with non-zero elements, in this case, $m_0 = 2$. All the rows below the first $m_0$ non-zero rows will contain only zeros and reflect the dependencies in the network. The objective of our algorithm is to generate $L_0$ matrix, which is done by row operations on $R$ matrix to first convert all diagonal elements to unity by dividing each row with a non-zero diagonal element by its value yielding a modified matrix $R'^*$ . Performing a Gauss-Jordan reduction to eliminate non-zero values above the diagonal of $R'^*$ gives $L_0$ as a submatrix of $R'^*$. This form can be matched to that described by Equation (13), where

\[
M \text{ corresponds to the transpose of } L_0. \text{ The } L_0 \text{ matrix is then obtained as the transpose of } M
\]

\[
R'^* = \begin{bmatrix}
1.0 & -0.5 & -1.0 & -0.5 \\
0.0 & 1.0 & 0.0 & -1.0 \\
0.0 & 0.0 & 0.0 & 0.0
\end{bmatrix}, \quad L_0 = \begin{bmatrix}
-1.0 & 0.0 \\
-1.0 & -1.0
\end{bmatrix}.
\]

The new order of species can be obtained by permuting the original species order with the permutation matrix $P$, and noting that the first $m_0$ species will comprise the independent species while the remaining ones will be dependent species. In our example, this new order is $[ES, S1, E, S2]$, with ES and S1 being independent species and E and S2 being dependent species. The conserved cycles can now be obtained as a product of the conservation matrix with the reordered species list as

\[
T = \begin{bmatrix}
1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1
\end{bmatrix} \begin{bmatrix}
ES \\
S1 \\
E \\
S2
\end{bmatrix} = \begin{bmatrix}
ES + E \\
ES + S1 + S2
\end{bmatrix}.
\]

8 COMPARISON TO LU DECOMPOSITION

The LU decomposition is a very well-known and standard numerical scheme employed to separate a matrix $A$ into a lower triangular matrix $L$ and an upper triangular matrix $U$. The LU decomposition is in fact used by PySCeS (Olivier et al., 2005) and COPASI (http://www.copasi.org) to factorize the stoichiometry matrix for structural and conservation analysis. LAPACK (Anderson et al., 1995) contains methods for obtaining the LU decomposition using both partial or full pivoting. Availability of these methods makes it feasible to apply the LU decomposition to conservation analysis in a straightforward manner, particularly for the method that uses partial pivoting. However, the method that employs full pivoting can be used only with square matrices. To overcome this restriction, we modify the stoichiometric matrix that is used as an input to the full pivoting method by adding additional zero rows or columns to convert it into a square matrix. We implemented both these methods using the CLAPACK library to compare the robustness and performance of the Householder QR factorization method. In the following, we briefly outline the implementation of the LU decomposition using partial as well as full pivoting.

The LU decomposition using partial pivoting of a general $m \times n$ matrix $A$ is given by $A = PLU$, where $L$ is an $m \times \min(m, n)$ lower triangular matrix, $U$ is a $\min(m, n) \times n$ upper triangular matrix and $P$ is an $m \times m$ matrix of row interchanges. While the partial pivoting method generates the $L$ and $U$ matrices, linear dependencies in the rows of $A$ are reflected by some of the diagonal elements of $U$ being zero. The primary shortcoming of partial pivoting is that
when LAPACK encounters a singular value on the diagonal of $U$, the algorithm is stopped and has to be restarted after the zero is exchanged with a non-zero value. An efficient search locates the positions of the zero diagonal elements and rearranges the columns such that all the non-zero diagonal elements are moved to the top of the $U$ matrix. It should be noted that this reordering has to be applied to columns of the transpose of the stoichiometric matrix as well as the order of the species. Once this has been done, the LU decomposition method has to be called again to complete the process that was interrupted by the existence of a zero on the diagonal element. When applied to the transpose of the stoichiometric matrix, the equivalent LU decomposition is given by $N^T = PL_U$, where as before, the subscript $t$ is indicative of matrices tracing their origin to the transpose of the stoichiometric matrix. If we denote by $(N^T)^t$ the stoichiometric matrix whose columns have been rearranged as described above, the second LU decomposition can be written as $(N^T)^t = P_tL_t^tU_t^t$, where $L_t^t$ is the new lower triangular matrix, $U_t^t$ is the new upper triangular matrix and $P_t^t$ is the new row reordering matrix. The matrix $U_t^t$ should now be reducible by Gauss–Jordan elimination of the elements above the diagonal to the form of Equation (13). The $L_0$ matrix can then be extracted as a submatrix of the Gauss–Jordan reduced $U_t^t$ matrix. It can be seen from Table 1 that the implementation of LU decomposition using partial pivoting generates nearly as many conservation laws as the QR factorization and is faster than the Householder QR, as can be seen from Table 2. However, for large network models (iJR904 and iND750), these conservation laws fail the validity tests described in the following section indicating the lack of reliability in using LU decomposition using partial pivoting on such large systems.

In the case of LU decomposition with full pivoting, an additional permutation matrix $Q$ containing column exchanges is used to increase the numerical stability of the method. The resulting decomposition can be written as $A = PL_UQ$. The advantage of using the full pivoting lies in the fact that the computation need be done only once. Further, the structure of the decomposition renders the method to a form that is similar to the Householder method, in that the column permutation matrix $Q$ in LU decomposition with full pivoting plays the same role as the transpose of the permutation matrix $P$ in the Householder method. We also infer that this gives the Householder method a marginal advantage in computational terms, since only one permutation matrix has to be computed. The remaining steps that are needed to compute the $L_0$ matrix from the $U$ matrix are similar to that followed for reducing the $R$ matrix in Equation (12) to the Gauss–Jordan reduced form in Equation (13). The $L_0$ matrix can then be obtained as $L_0 = M^T$ and the conservation laws constructed using the relation $\Gamma = [-L_0I]$. We can indeed show that the conservation laws obtained by the Householder decomposition and the LU decomposition with full pivoting are equivalent, in that both pass the validation tests detailed in Section 9. Consequently, we now have two independent methods to verify the conservation laws for large biochemical networks, with the Householder QR method being slightly faster than the full pivoted LU decomposition method.

### 9 TESTING CONSERVATION LAWS

The conservation laws generated by the algorithm must be tested to ensure that they are valid. Our algorithm therefore includes five tests to validate the conservation laws. The first test checks if the conserved sums are indeed constant values. This is done by observing that the conserved sums are given by $T = TS$. Their rate of change, $dT/dt$ must be zero if $T$ is a constant vector. This is equivalent to

$$\frac{dT}{dt} = \frac{d}{dt}(TS) = \Gamma Nv = 0. \quad (18)$$

The second test checks if the rank obtained by QR factorization of the stoichiometric matrix $N$ is the same as that obtained by counting the non-zero singular values obtained from an SVD of $N$. If the rank computed by the two methods is different, it is evident that the number of conserved cycles has been miscalculated.

The third test checks the rank of $N_R$ matrix. Since the $N_R$ matrix corresponds to the contribution of all the independent species in the network, it must have a full rank. Therefore, one can compare the rank of $N_R$ using QR factorization to that obtained by SVD. If these ranks are full and match, then the conservation laws are correct.

The fourth test involves the computation of the eigenvalues of a submatrix obtained from the QR factorization of $N$. We note that the orthogonal matrix $Q$ obtained after factorizing the reordered stoichiometric matrix can be partitioned into four submatrices as described in Equation (10). If the stoichiometric matrix $N$ has been reordered such that the first $m_0$ rows correspond to independent species, and the remaining $(m - m_0)$ rows correspond to dependent species, it can be shown that $Q_{11}$ is a non-singular and invertible matrix. We can therefore compute its rank by counting the number of non-zero eigenvalues of $Q_{11}$. If $Q_{11}$ is of full rank, i.e. if $m_0 = \text{rank}(Q_{11})$, the conservation laws must be correct.

The fifth test evaluates the $L_0$ matrix as the product of the submatrices of $Q$. As observed earlier, $L_0 = Q_{21} Q_{11}^{-1}$. The matrix obtained by multiplying $Q_{21}$ and the inverse of $Q_{11}$ are compared with the $L_0$ matrix obtained directly by the Householder method. The test can then be deemed successful if both matrices match.

The results for these validity tests depend on the size of the model and the method employed to generate the conservation laws. The numerical errors inherent in the algorithm are highest in the case of LU decomposition with partial pivoting, leading to the generation of conservation laws that do not pass the validity tests. This can be seen in Table 1. On the other hand, the Householder QR method developed in this paper and LU decomposition with full pivoting (used by PySCeS) generate the same number of conservation laws, both of which pass the validity tests.

### 10 SOFTWARE IMPLEMENTATION

The software developed for the purpose of demonstrating the capabilities of the Householder QR method is a part of the latest
11 APPLICATION TO SBML TRANSLATION

Many biochemical models are now available in SBML (Hucka et al., 2003) format, which makes it feasible to share and disseminate information regarding new developments (http://www.biomodels.net). However, from a simulation point of view, SBML has to be parsed to obtain information about the species, reactions and rate-laws in order to create a model that can be simulated. Currently existing software tools make it feasible to translate SBML into MATLAB (http://www.mathworks.com/) code that can be run in the MATLAB simulation environment. For small models, this is quite convenient as the time evolution of various species can be graphed easily. In the case of large models however, MATLAB turns out to be very slow as the number of species and reactions increase many fold.

Faced with the slow simulation in MATLAB, we have developed an alternative to this by developing tools that can translate SBML into higher languages such as C, C# and Java, where such simulation can be carried out very quickly for large systems. For instance, a 20-fold speedup in simulation time over MATLAB was achieved using the SBML to C Translator. The translators also provide methods that allow computation of the reduced Jacobian of the system that is very useful for carrying out other analyzes.

The conservation analysis approach presented in this paper has now been integrated into all our SBML translators. This allows the users to translate models in SBML to a more simulation friendly language such as C or C# or Java. The SBML translators call the Structural Analysis methods to obtain a list of the independent and dependent species. The dependent species are then eliminated from the differential equations that have to be simulated using the conservation laws that are generated by the Structural Analysis tool. This also allows for computation of the reduced Jacobian that is non-singular. The users can therefore easily convert existing model information contained in SBML format to other languages. This has been illustrated in Figure 3.

Another example of the application of the Structural Analysis module is that of Oscill8 (http://sourceforge.net/projects/oscill8), a Bifurcation Analysis package that utilizes AUTO (Doedel et al., 1991). This package interfaces with Systems Biology Workbench to provide a seamless interface that allows users to study the dynamics of their models. This is done by translating the biochemical model described in SBML to XPP format, which defines the reduced system in terms of ordinary differential equations and their associated initial conditions. This XPP translated model is then used by Oscill8 to carry out bifurcation studies.

12 RESULTS AND DISCUSSION

A Householder method-based approach for conservation Analysis has been presented in this paper. We have shown that it is a robust method capable of finding the correct conserved cycles for some very large biochemical models. This method uses the CLAPACK SBW library for dense matrices, and a sparse matrix-based solution is under development for very large systems.

The Householder-based method along with the LU decomposition using full pivoting appear to be the most promising of the currently known methods for conservation analysis of large biochemical networks. In addition to extracting the conservation laws, the software developed for this purpose also provides five tests to verify if the conservation laws satisfy required conditions.

Biochemical models involve reactions between various species, and often the action of many species is restricted to just a few reactions. Similarly, some reactions involve very few reactants and products. As a result, the stoichiometric matrix tends to be sparsely populated, with the sparseness increasing with the size of the network. Most analysis methods tend to treat the stoichiometric matrix as a dense matrix where the computations are carried out on all elements of the matrix, even if they are zero. This makes the algorithm slow and at the same time occupies computer memory. A sparse matrix scheme for representing stoichiometric matrices, along with numerical routines capable of performing analysis on sparse matrices will be beneficial in increasing the computational speed and efficiency in future.

Typical sparse matrix representation methods keep a list of the non-zero values, along with their row and column index locations. This is a compact representation that saves memory, but is not efficient from a computational perspective. For most matrix operations that involve rowwise or columnwise operations, a lot of time would be expended on locating the next non-zero row or column element. We are therefore implementing a sparse matrix scheme that not only keeps a list of non-zero values and their column indices, but also assigns pointers to the previous and next non-zero elements, and the previous and next column non-zero elements.
This is called a doubly linked data structure and allows for faster element search. We believe this approach will lead to the development of sparse matrix-based analysis tools that will make it possible to study the dynamics of very large biochemical networks.

In this paper, we have not addressed the issue of computing conservation laws where the number of negative terms is minimized (Sauro, 1993). However, we refer the reader to the paper by Kholodenko et al. (1995) which briefly discusses ordering rules to minimize the number of negative terms.

The numerical method developed herein makes it feasible to attempt the more complex problem of simulation of biochemical systems with varied time scales. Typically, these large systems are stiff due to the occurrence of several fast processes which require small step sizes for integrating the solution. Park (1974) first addressed the issue of developing simulators that took into account the temporal hierarchical structure of metabolic networks by splitting the system into slow and fast variables (Heinrich and Schuster, 1996). Such a split assumes that the system moves quickly onto a manifold which is slow. Hence, if such a manifold is found, then the task of integrating the system into slow and fast variables (Heinrich and Schuster, 1996) simplifies the task of integrating the system into slow and fast variables (Heinrich and Schuster, 1996). Such a split assumes that the system moves quickly onto a manifold which is slow. Hence, if such a manifold is found, then the task of integrating the system into slow and fast variables (Heinrich and Schuster, 1996).

Recently, efficient methods have been developed, which use these intrinsic low-dimensional manifolds (ILDM) to solve large chemical systems, for example in the field of combustion (Mass and Pope, 1992; Nafe and Mass, 2002). In another publication, we will describe the implementation of these methods in conjunction with the conservation analysis approach that we have elaborated upon in this paper.

ACKNOWLEDGEMENTS

R.V. would like to acknowledge Brett Olivier for assistance with converting large SBML models to PySCeS format, Emery Conrad for testing the conservation Analysis tool with Oscil8, Anastasia Deckard for generating displays of large biochemical network models and Frank Bergmann for assistance with software development. We would also like to thank the reviewers of this paper for bringing to our attention the fact that PySCeS employs LU decomposition using full pivoting. R.R.V. and V.C are grateful to the generous support from the US DOE G1 Program and the DARPA/IPTO BioCOMP program, contract number MIPR 03-M296-01 respectively.

Conflict of Interest: none declared.

REFERENCES


