

NYU BioWave & NYU BioSim: Automating Analysis of BioChemical Pathways *

MARCO ANTONIOTTI¹, PAOLO EMILIO BARBANO^{1,3}, WILLIAM CASEY¹, JIAWU FENG¹,
MARC REJALI¹, MARINA SPIVAK¹, NADIA UGEL¹, AND
BUD MISHRA^{1,2†}

¹ Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY, USA 10012.

² Cold Spring Harbor Lab, 1 Bungtown Road, Cold Spring Harbor, NY, USA 11724.

³ Department of Mathematics, Yale University, New Haven, CT, USA

February 7, 2004

Abstract

This paper describes a set of novel tools for analyzing trajectories of bio-chemical pathways, where these trajectories are obtained either through in silico simulations or through in vitro or in vivo time-course data. In particular, we describe two interesting tools—NYU BioSim and NYU BioWave, within a more general Simpathica system—that store, analyze and group time-series representations of bio-chemical trajectories by using a multi-resolution time-frequency analysis for optimal basis selection. We illustrate, by examples, how it can be used to understand the behavior of a family of artificial biological circuits. We also show how these tools interact with a model-checking system to present qualitative distinctions among the groups within the family of biological circuits or among the different multi-modal behaviors of a single pathway.

requiring consilience of elegant ideas and concepts from applied mathematics, theoretical computer science, logic and physical modeling. The impulse has come from better understanding of processes involved at molecular level, technology at meso- and nano-scale, ability to perform high through-put experiments and vast amount of genomic and proteomic data that can now be generated and made publicly available for processing. In response to these challenges, a group of scientists and mathematicians belonging to NYU/COURANT BIOINFORMATICS GROUP has concentrated its collective attention on these questions.

An accelerating impulse to this group’s work was provided by the DARPA’s BioCOMP/BIOSPICE project involving several external investigators as well. As a part of this research effort, we have been creating computational tools (e.g., Simpathica, NYU BioSim, NYU BioWave and XS-System—subject of this paper), integrating these tools with the other tools in the larger effort (www.biospice.org), and participating in the design of the systems, languages and experiments involved in this effort.

1 SOME PRELIMINARY REMARKS

Understanding biology by modeling cellular processes and genome evolution has emerged as a challenging new area: “systems biology.” Sitting at the interface of mathematics and biology, this subject aims to address many questions

The group focuses on four areas of research. (a) Biochemical Process Theory, (b) Evolutionary Processes, Genomes and Pathway Models, (c) Advanced Tool Architectures and (d) Experimental Research. The main emphasis is naturally placed on providing biologists and biotechnologists with the capability to analyze large and complex biological systems and devise intelligent experiments without being forced to deal with the mathematical details and complexity of the system. NYU/COURANT BIOINFORMATICS GROUP has developed and implemented a computational system, Simpathica, which allows users to construct and rigorously analyze models of biochemical pathways composed out of a set of

*The work reported in this paper was supported by grants from DARPA’s BioCOMP project (Title: “Algorithmic Tools and Computational Frameworks for Cell Informatics”) and AFRL contract (contract #: F30602-01-2-0556). Additional support was provided by NSF’s Qubic program, HHMI biomedical support research grant, the US department of Energy, the US air force, National Institutes of Health and New York State Office of Science, Technology & Academic Research.

[†]To whom correspondence should be addressed. E-mail: mishra@nyu.edu

basic reactions—such as *reversible reaction*, *synthesis*, *degradation*, *reaction modulated by enzymes and coenzymes*, *multimerization*, etc. Because of the fundamental nature of these basic building blocks, it is relatively easy to connect *Simpathica*, through a translator, to other public pathway databases — e.g., NCI CGAP (Cancer Genome Anatomy Project), KEGG, Biocarta, Biocyc, etc. *Simpathica* is able to construct a rigorous mathematical description of these pathways through PDE’s, ODE’s, and SDE’s; create a qualitative model (Kripke structure or hybrid automata) efficiently; and compose these models hierarchically and reason about the system’s behavior in a propositional branching time temporal logic. Thus, *Simpathica* is powerful enough to deal with large biochemical systems, disease models, or models dealing with a large family of cell lines, and mutants.

Furthermore, *Simpathica* has an “easy-to-use” structure that “hides” all the mathematical details: users create models of the biochemical pathway diagrammatically (or download existing models that are further modified and then composed) and navigate through the analysis tools either by visual inspection of the trajectory or by engaging in a dialogue with *Simpathica* by proposing various hypotheses that *Simpathica* either ascertains or refutes—when *Simpathica* refutes a hypothesis it provides a “counter-example” to the user. Moreover, because of qualitative nature of the analysis, often *Simpathica* can analyze a system convincingly even when it does not have access to the full set of kinetic parameters operating *in vivo*.

Simpathica can deal with traces (time course data) that are the product of wet-lab experiments or computer simulations. *Simpathica* manipulates these traces with a variety of techniques and tools: standard visualization tools, exhaustive “queries” expressed with a branching time propositional temporal logic formalism, clustering and pattern matching using multiresolution time-frequency techniques.

Generally speaking, starting as an input a trace of a biochemical pathway, (i.e. a time-indexed sequence of state vectors representing a numerical simulation of the pathway), *Simpathica* can perform the following operations.

- *Simpathica* answers complex questions involving several variables about the behavior of the system. To this end we defined a query language based on temporal logic formalism. Thus we can, formulate queries like

```
eventually(not always(LacI < 1.3)
or always(LacI > 4.0)).
```

In the above example, the query expresses the fact that the value of the ‘LacI’ variable “oscillates” between the two values of 1.3 and 4.0. The system being analyzed is the *repressilator* system of Elowitz and Leibler.

The analysis tool provides counter examples when input query fails to hold true or restricts the conditions under which the query can be satisfied.

- *Simpathica* stores traces in a database and allows easy search and manipulation of traces in this format. The analysis tools allow these traces to be further examined to extract interesting properties of the bio-chemical pathway. *Simpathica* contains a prototype subsystem (called NYU BioSim) as its main simulation database.
- *Simpathica* classifies several traces (either from a single experiment or from different ones) according to features discernible in their time and frequency domains. Multiresolution time-frequency techniques can be used to group several traces according to their features: steps, decreases, increases, and even more complex features, such as, memory. *Simpathica* contains a prototype subsystem (called NYU BioWave), which implements these classification procedures using Matlab.
- With these tools, *Simpathica* provides an environment to suggest plausible hypotheses and then, refute or validate these hypotheses with experimental analysis of time-course evolution. It also allows investigating conditions or perturbations under which a metabolic pathway may modify its behavior to produce a desired effect (an instance of a control engineering problem).

2 MATHEMATICAL MODELS AND TRAJECTORY GENERATION

In *Simpathica*, biochemical reactions are modeled with sets of differential equations. Each reaction is thought of as a module and belongs to one of many types: *reversible reactions*, *synthesis*, *degradation*, and *reactions modulated by enzymes and co-enzymes* or other reactions satisfying certain *stoichiometric constraints*. If the stochastics in these reactions are ignored (i.e., mass-action models), each of these reactions can be described by a first order algebraic differential equation whose coefficients and degrees are determined by a set of thermodynamic parameters. As an example, reaction modulated by an enzyme leads to the classical Michaelis-Menten’s formulation of reaction speed as essentially differential equations for the rate of change of the product of an enzymatic reaction. The parameters of such an equation are the constants K_m (Michaelis-Menten Constant) and V_{max} (maximum velocity of a reaction). In a simple formulation, such as in S-system [Voi91, Voi00], this approach provides a convenient way of describing a biochemical pathway as a composition of several primitive reaction

modules and then automatically translating them into a set of ODE's with additional algebraic constraints. **Simpathica** and XS-system [Mis02, APP⁺03] (an extension of the basic S-System) retains this modular structure while allowing for a far richer set of modules.

Canonical Forms. A set of differential equations in XS-system can always be rewritten (recast) in special canonical forms by purely algebraic transformations and further inclusions of a set of algebraic constraint equations. Canonical forms have several advantages over more general forms of equations, since they can be more easily manipulated, integrated and interpreted in mathematical terms.

An XS-system is simply a list of expressions describing the rate of change of a given quantity in a model (say the concentration of a compound), plus a set of equations describing some constraints on the relationships among some of the parameters characterizing the model. Each of the expressions describing a rate has a very simple form as well: it is simply a difference between two algebraic power-products (or monomials) one representing synthesis and the other, dissociation. More formally we have the following: An XS-system is defined by a set of pairs of equations (a rate equation and a constraint equation)

$$\begin{aligned} \dot{X}_i &= \alpha_i X_1^{g_{1i}} X_2^{g_{2i}} \dots X_n^{g_{ni}} - \beta_i X_1^{h_{1i}} X_2^{h_{2i}} \dots X_n^{h_{ni}} \\ 0 &= (a_{1j} X_1^{c_{11j}} X_2^{c_{12j}} \dots X_n^{c_{1nj}}) \\ &\quad + (a_{2j} X_1^{c_{21j}} X_2^{c_{22j}} \dots X_n^{c_{2nj}}) \\ &\quad + \dots + (a_{mj} X_1^{c_{m1j}} X_2^{c_{m2j}} \dots X_n^{c_{mnj}}) \end{aligned}$$

with index variables, i ranging from 1 to n , and j , from 1 to k . This formalism describes an XS-system with n equations and k constraints. An XS-system can be interpreted as the representation of a set of flows of reactants within a network of reactions [Voi00] and thus describes how to algorithmically translate a graphical rendition of such reaction networks into the equations in a canonical form. Our XS-system formulation naturally captures these steps in a computer-assisted translation, which had been traditionally carried out by a manual manipulation; see [Voi00].

The XS-system formulation makes one more distinction between dependent and independent variables. Independent variables represent environmental conditions which influence the behavior of the system but which do not influence themselves in return. Dependent variables are all the others. Of course, to complete the description of the system it is necessary to specify all the *rate constants* (α 's and β 's) and the *kinetic orders* (g 's, h 's, and c 's) of each equation and constraint.

Once such a representation is obtained, behavior of the system can be analyzed by examining the temporal relations between the independent and dependent variables in terms

of the sets of trajectories (traces) as the initial conditions and parameters vary over their possible realistic values. The sets of tools presented here provide an automated approach to derive the equations for the biochemical pathways, numerically simulate them to create trajectories as time-series traces, store and catalogue these traces in a database and analyze and classify these functional data to gain insight into the biological function of the pathway.

3 TRAJECTORIES STORAGE IN NYU BIOSIM

NYU BioSim is a database system for storing time-indexed simulation data. The need for such a system arose from the fractious state of affairs met by several researchers within the NYU/COURANT BIOINFORMATICS GROUP and outside it—namely, in the larger BIOSPICE community.

Time-indexed (or time-course, time-series) data is being generated by many researchers and they always appear in the format

$$\langle t, v_1, v_2, \dots, v_k \rangle_i, \text{ for } i = 0, \dots, N,$$

decorated with some “meta” information, such as the name of the quantities being measured and the circumstances of the “experiment”. We found that managing this kind of data in a more organized way is key to making sure that our research results are easily reproducible and analyzable, especially by third party laboratories.

Thus, we decided to build a simple yet versatile, centralized facility to ease the storage, retrieval, and above all, classification of time-indexed data sets: NYU BioSim.

The system has a three-tier architecture insuring scalability. A **Postgresql** relational database management system forms the back-end tier. The middle application tier comprises Java servlets and supporting modules that respond to client requests and interact with the database. The front-end is a Java application that provides an easy and intuitive GUI (graphical user interface). The GUI communicates with the server side using an XML data exchange format over HTTP. The architecture is illustrated in Figure 1.

The system is accessible to anyone with an internet connection¹. Users with IDs and passwords can save, edit and retrieve private data. Other users can log on as a **guest** and view and retrieve public data. The **login** screen is shown in Figure 2.

The system allows controlled access to data so that only users with the correct authorization can view private data.

¹See <http://bioinformatics.cat.nyu.edu/nyumad> for information on how to download and use the client GUI application.

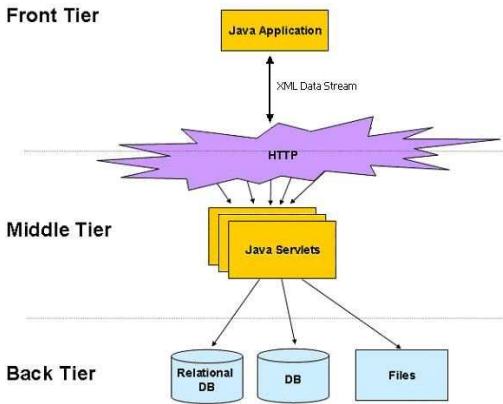


Figure 1: The three-tiered architecture of NYU BioSim.

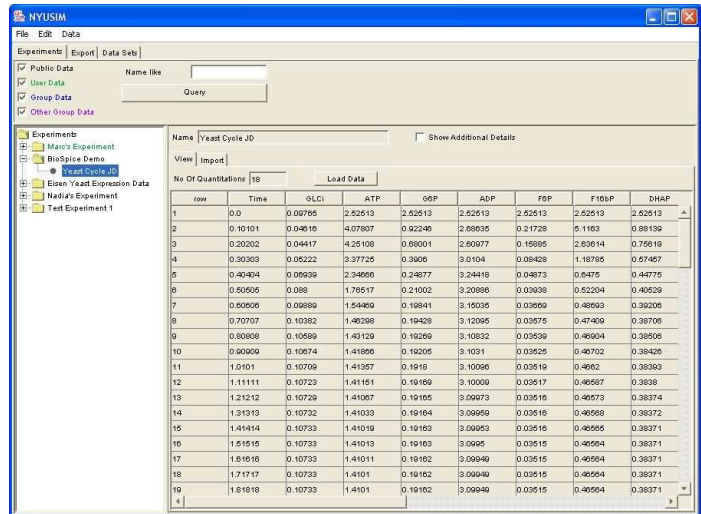


Figure 3: NYU BioSim data set.

Each dataset has an ownership that determines its visibility. Collaborating groups can allow shared visibility of the data between their groups. After publication data can be made publicly available with a simple command. Public data can be viewed by all users, including guest users.

The system stores a set of simulation trace data as a matrix, each column representing a simulated variable and each row representing a time point. Simulation data sets (matrices) are grouped under an experiment. Users create experiments, and for each experiment they can generate and store several sets of simulation data. Figure 3 shows a view of one such data set.

The GUI makes the importing of new data easy. New data sets are imported in to the system by cutting and pasting in to an importing area or by loading from a file. After importing data, synthetic data sets can be created by combining columns from different but compatible matrices. Data can be exported to the system clipboard from all the screens where matrix data is loaded or viewed, providing very flexible and efficient data retrieval for further analysis. There is a custom 'Export' screen where any combination of compatible columns can be exported.

The security model of the system controls visibility and read/write access to the data. Each user belongs to a primary group which gives them read access to all the data belonging to members of that group. An administrator tool is used to set and edit a user's write access and additional access rights to data from other groups.

For viewing data, users have the flexibility to restrict data query to data categories of interest. This will be a useful feature as the number of experiments and data sets increases.



Figure 2: NYU BioSim login screen. The NYU BioSim user interface is architecture independent and it will work on any platform that supports a Java virtual machine.

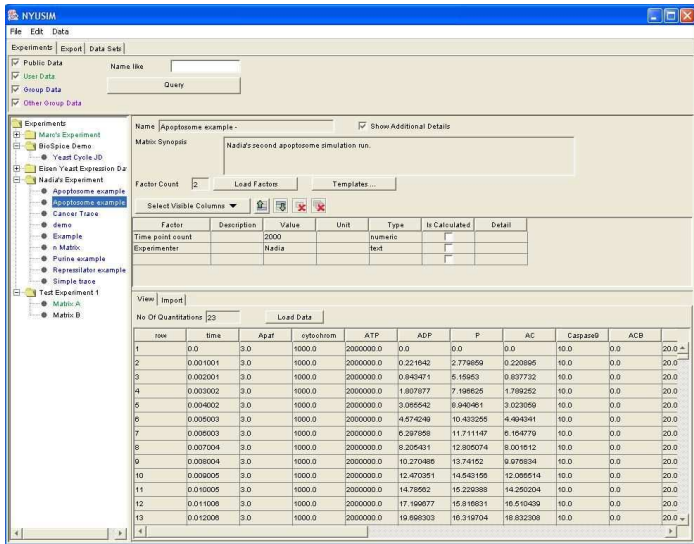


Figure 4: A data set with two factors and a synopsis.

The query panel can be seen on the two figures above. There are four major data categories.

1. *Public Data*: visible to all users including 'guest' users.
2. *User Data*: the user's private data, visible only to other members of the same group.
3. *Group Data*: data from other members in the same group as the user.
4. *Other Group Data*: data from other groups giving the user access rights.

Collaborating groups that share data will see the data from other groups under the '*Other Group Data*' category. In the tree view of the data hierarchy, the different data categories are color-coded for easy identification. Furthermore, the data query can be restricted to experiments with names matching a given pattern.

In addition to basic simulation data it is possible to store associated data such as experimental factors and parameters as well as free format descriptive text for each experiment or data set. If there are common sets of factor and parameter data, a template of such factors can be created for easy input. Figure 4 shows a data set with two factors and a very brief synopsis.

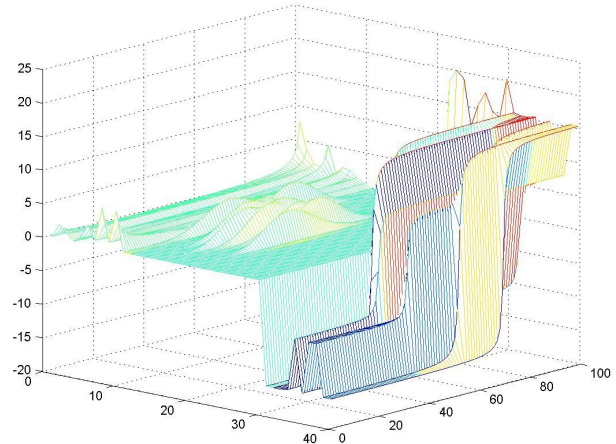


Figure 5: Simple test case used to evaluate NYU BioWave: 30 β functions evaluated with different parameters, and 10 *step* functions with different shifts, steepness and amplitude.

4 TIME-FREQUENCY ANALYSIS WITHIN NYU BIOWAVE

Many biological experiments (especially *in silico* experiments) produce *time course data* which can be analyzed both in time and frequency domains to extract interesting functional properties. To this end we have constructed NYU BioWave, a tool that can find similarities in the 'shape' of time course data, that is, it can easily group together measurements of different quantities based on their time-course behavior. As an example, it can group together all trajectories that present a 'step' feature, thus easing the detection of relationships among observed variables. Moreover, it can do so across several datasets (e.g. datasets corresponding to different values of controlled parameters.)

The mathematical theory behind the NYU BioWave tool is primarily based upon *Multiresolution Time-Frequency Analysis* through *Wavelet Decompositions* [Ma199]. We will describe the overall structure of our application in Section 4.1. In Figures 5 and 6 we show a simple and artificial test case used to validate NYU BioWave capabilities, and the NYU BioWave user interface (built in Matlab).

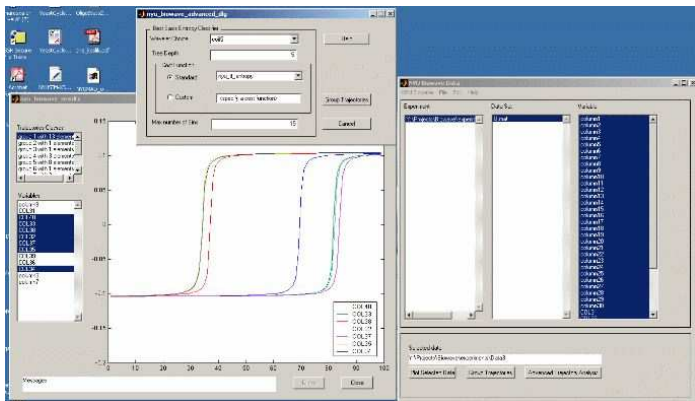


Figure 6: A view of the NYU BioWave user interface. There are three windows visible. In the background there is the dialog showing the connection to NYU BioSim, in the foreground there are the two windows that constitute the “classifier inspection tool.” The group comprising the *step* functions is being reviewed (the functions are normalized before being plotted).

4.1 USING A MULTISCALE OPTIMAL BASIS SELECTION ALGORITHM TO CLASSIFY TRAJECTORIES

NYU BioWave utilizes a multiscale basis selection algorithm. The first example in this class of algorithms, the *best basis algorithm* can be found in [CW92]. There, given bi-orthogonal wavelet filter denoted by $[v, w]$, the best basis algorithm defines a method for searching a subset of $O(M)$ (the set of orthogonal transformations in \mathbb{R}^M). $O(M)$ is generated by wavelet filter trees $[v, w]$, and has a number of interesting mathematical properties, which we do not discuss here (again, cfr. [CW92]). We denote the subset analyzed by the algorithm as $K[v, w] \subset O(M)$. The best-basis algorithm searches $K[v, w]$ by means of a heuristic tree pruning algorithm.

NYU BioWave implements a computational scheme to analyze arbitrary continuous function $\alpha : \mathbb{R}^M \rightarrow \mathbb{R}$. Given a wavelet filter $[v, w]$ and a continuous function α , NYU BioWave defines a method for searching a subset of $O(M)$ that uses a tree pruning algorithm whose operation is governed by the function α . The original best-basis algorithm is then an instance (with α being the entropy function) of the algorithm implemented in NYU BioWave.

Trajectory Classification

NYU BioWave eventually associates a ‘score’ $s_i \in \mathbb{R}$ to each trajectory f_i examined, with $i = 0 \dots n$.

Currently, the ‘score’ is a value derived from the *entropy* of the trajectory. The set of scores is simply $S = \langle s_i \rangle$. These scores are then partitioned in groups, according to the characteristics of their distributions. At present, NYU BioWave implements a simple grouping scheme that optimizes *gaps* between the groups. The scheme is based on the computation of a “moving average” $\hat{\mu}$ and relative standard deviation $\hat{\sigma}$ of the “distances” $D_S = \langle s_{i+1} - s_i \rangle$ between the scores. Two scores s_i and s_j are grouped separately if $|s_j - s_i| > \hat{\mu} + 2\hat{\sigma}$. Of course, this method of clustering entropy scores is rather coarse and arbitrary and requires further research. However, we note that this approach works well when there is a known correlation among the f_i ’s (as is the case with the example described in Section 5).

An alternative and a more sophisticated way to assign a score to each trajectory would be to compute the set $\{\epsilon_{ij}\}$ defined as the “entropy of the coefficients of the representation of f_i , with respect to the best basis computed for f_j .” We could then group f_i and f_j together, based on $\|\epsilon_{ij} - \epsilon_{ji}\| \leq \kappa$, for a given parameter κ . In other words, we consider a pair of functions similar, when they are ‘close’ with respect to their representation in terms of the optimal basis associated to the function².

Finally, we note that, this clustering problem is quite difficult to solve in a complete general and more sophisticated way, and we will explore it in more detail in a different setting.

5 BIOLOGICAL CIRCUIT OF GUET ET AL.

As a rather simple example of how NYU BioWave and NYU BioSim may be used in analyzing biological systems, we will focus on a “bio-circuit” originally designed by Guet and others [GEHL02].

The original motivation for designing such a family of synthetic networks by combinatorial variations of the network topology were given as follows [GEHL02]: “A central problem in biology is determining how genes interact as parts of functional networks. Creation and analysis of synthetic networks, composed of well-characterized genetic elements, provide a framework for theoretical modeling. ... Combinatorial synthesis provides an alternative approach for studying biological networks, as well as an efficient method for producing diverse phenotypes *in vivo*.” Nonetheless, lack of efficient tools for modeling and analysis of such synthetic networks has hindered many possible applications of these

²We also note that the criteria we described is not symmetric. We will describe the detail of our approach in a different setting.

networks. Clearly, with appropriate tools, one could foresee applications where millions of randomly generated networks could be screened for selection of primitive circuits with specific properties (robustness, immunity to noise, etc.), or as building blocks of larger circuits with specific temporal properties, or even as scaffold structures for measuring kinetic parameters of a component as it operates *in vivo*. Here, we suggest that NYU BioSim and NYU BioWave and their planned software progenies respond to these demands quite well.

In the scheme created by Guet and colleagues, they used a combinatorial method to generate a library of networks with varying connectivity and implemented them as plasmids capable of transfecting *Escherichia coli*. These networks were composed of genes encoding the transcriptional regulators LACI, TET, and λ CI, as well as the corresponding promoters. Although the networks had time-varying output trajectories for a fixed input and implemented sequential circuits, Guet et al. characterized their phenotypic behaviors as resembling binary logical/combinatorial circuits, with two chemical “inputs” and a fluorescent protein “output.” Nevertheless, the biological experiments indicated a rich and diverse set of functions dependent on network connectivity and raised questions about how to design appropriate computational tools to analyze them.

In this paper [GEHL02], the authors generated a combinatorial library composed of a small set of transcriptional regulatory genes and their corresponding promoters and varied their connectivity in a combinatorially exhaustive manner. They chose genes of three well-characterized prokaryotic transcriptional regulators: *Lac*, *Tet*, and λ *cI*. The binding state of *LacI* and *TetR* can be changed with the small molecule inducers, isopropyl b-D-thiogalactopyranoside (IPTG) and anhydrotetracycline (ATC), respectively. In addition, they also selected five promoters regulated by these proteins (i.e. LAC, TET, and λ CI), which span a rather broad range of regulatory characteristics—e.g., repression, activation, leakiness, and strength. Two of the promoters are repressed by LAC (to be referred to as PL1 and PL2), one is repressed by TET (to be referred to as PT), and finally, the last two are regulated by λ CI, one positively ($P\lambda_+$) and one negatively ($P\lambda_-$). Their genetic assembly scheme ensured that each network in the library has the following structure: P_i -*lac*- P_j - λ *ci*- P_k -*tet*, where each P_i , P_j , and $P_k \in \{PL1, PL2, PT, P\lambda_+, P\lambda_-\}$ is implemented as any of the five promoters. Thus, the regulatory genes on each plasmid interact (i.e., activate or repress) with one another, generating networks with diverse connectivities. A separate plasmid consisting of a reporter *gfp* and repressed by λ *ci* is used to measure the biological activity

of the synthetic network through the fluorescence of GFP.

In this paper, we will model all possible $5^3 = 125$ different networks and by examining their trajectories group them into various classes and examine how well this grouping coincides with the others based on topology. Since the networks constructed this way encompass a wide range of motifs (including negative and positive feedback loops, oscillators, and toggle switches) they present an interesting family of trajectories to NYU BioWave.

In summary the system to be analyzed consists of the following:

1. There are combinations of four genes: *lac*, λ *ci*, *tet* and *gfp*, of which the first three interact with each other by pair-wise activation or repression and the last one (*gfp*) is used as an output. The corresponding proteins are denoted as LAC, λ CI, TET and GFP. Their concentrations will be indicated by the notation $[x]$ (e.g., $[lac]$ is the concentration of *lac*-mRNA and $[LAC]$ is the concentration of LAC-protein). The temporal rate of change of concentration will be denoted as $[\dot{x}]$.
2. The small molecule inducers IPTG and ATC act as the inputs to the system through their inactivation of the *lac* and *tet* genes, respectively.
3. There are five Operons: two LAC-based: PL1, PL2; two λ CI-based: $P\lambda_-$, $P\lambda_+$; one TET-based: PT.
4. Total $5^3 = 125$ different combinatorial circuits are possible. A circuit is denoted as P_i -*lac*- P_j - λ *ci*- P_k -*tet*, indicating that P_i determines the transcriptional state of *lac*; P_j determines the transcriptional state of λ *ci* and P_k determines the transcriptional state of *tet*.
5. For instance the circuit $P\lambda_+$ -*lac*-PL1- λ *ci*-PL1-*tet* has the following connections:
 - (a) *lac* is *activated* by λ CI.
 - (b) λ *ci* is *repressed* by LAC, and LAC is inactivated by IPTG.
 - (c) *tet* is *repressed* by LAC, and LAC is inactivated by IPTG.
 - (d) *gfp* is *repressed* by λ CI.

In our analysis we will make several simplifying assumptions: (1) All genes have similar time constants; (2) mRNA’s instantaneous concentration depends on the transcription process, its leakiness and its instability (i.e., how it degrades); (3) Protein’s instantaneous concentration depends on the translation process and its degradation. Their dynamic state-evolution equations can be written in terms of

two intrinsic parameters α (governing mRNA) and β (governing protein) as well as Hill-coefficient like terms (n and k), leakiness term (ρ) and saturation terms (θ).

If x denotes a gene and X its corresponding protein, we have the following equation for x 's transcription:

$$[\dot{x}] = -[x] + \alpha[\rho + f_x(\theta, [Y], [u_y])]$$

where

$$f_x(\theta, [Y], [u_y]) = \frac{1 + \theta[Y]^n + [u_y]^k}{1 + [Y]^n + [u_y]^k}.$$

In this equation, the transcription is activated or repressed by a protein Y and Y , itself is modulated by a small molecule u_y . Note that, for small values of $[u_y]$, f_x shows a sharp transition from a value of 1 (when $[Y] = 0$) to a value of θ (when $[Y] = \infty$), as Y increases. However, for large values of $[u_y]$, f_x remains at 1 (when $[u_y] = \infty$), thus inactivating the effect of Y .

Similarly, we have the following equation for X 's (corresponding protein) translation:

$$[\dot{X}] = -\beta([X] - [x]).$$

Going back to our example circuit $P_{\lambda+}lac-PL1-\lambda ci-PL1-tet$, we can write down in a straightforward manner the corresponding ODE's as shown below:

$$[\dot{lac}] = -[lac] + \alpha\rho + \alpha\frac{1 + \theta_a[\lambda CI]^n}{1 + [\lambda CI]^n}$$

$$[\dot{LAC}] = -\beta([LAC] - [lac])$$

$$[\lambda \dot{ci}] = -[\lambda ci] + \alpha\rho + \alpha\frac{1 + \theta_s[LAC]^n + [IPTG]^k}{1 + [LAC]^n + [IPTG]^k}$$

$$[\dot{\lambda CI}] = -\beta([\lambda CI] - [\lambda ci])$$

$$[\dot{tet}] = -[tet] + \alpha\rho + \alpha\frac{1 + \theta_s[LAC]^n + [IPTG]^k}{1 + [LAC]^n + [IPTG]^k}$$

$$[\dot{TET}] = -\beta([TET] - [tet])$$

$$[\dot{gfp}] = -[gfp] + \alpha\rho + \alpha\frac{1 + \theta_s[\lambda CI]^n}{1 + [\lambda CI]^n}$$

$$[\dot{GFP}] = -\beta([GFP] - [gfp])$$

Thus,

1. The first two equations model the fact that *lac* is *activated* by λCI .
2. The next two equations model the fact that λci is *repressed* by LAC, and LAC is inactivated by IPTG.

3. The next two equations model the fact that *tet* is *repressed* by LAC, and LAC is inactivated by IPTG.

4. The last two equations model the fact that *gfp* is *repressed* by λCI .

We used the following parameters and simulation functions:

$$[IPTG](t) = -\exp(-t)[IPTG](0)$$

$$[IPTG](0) = x_0 = 3$$

$$[ATC](t) = -\exp(-1.1t)[IPTG](0)$$

$$[ATC](0) = y_0 = 3$$

$$\alpha = 5$$

$$\beta = 1$$

$$\rho = 0.1$$

$$\theta_s = 0 \quad \text{implying suppression}$$

$$\theta_a = 2 \quad \text{implying amplification}$$

$$n = 2$$

$$k = 2,$$

and note that in our normalized equations, we have

- α = concentration of proteins per cell from *unrepressed* promoter
- $\alpha\rho$ = concentration of proteins per cell from *repressed* promoter
- β = protein : mRNA decay rate ratio
- n = Hill (cooperativity) coefficient of the repressor
- k = Hill (cooperativity) coefficient of the small molecule

5.1 Analysis

We ran simulations for each of the 125 circuits with the inputs listed in Table 1. The simulations were run using Matlab standard Ordinary Differential Equations integrators. In each run all 125 circuits were tested until a steady state was reached. The result was a set of 125 trajectories for each input pair (IPTG, ATC) (i.e. 4 sets). Two kinds of analysis were performed on the resulting sets of data: a *time-frequency* analysis using NYU BioWave and a classification of combinatorial circuits using Simpathica/XSSYS.

IPTG	ATC
0.0	0.0
0.0	3.0
3.0	0.0
3.0	3.0

Table 1: Initial concentrations of the *input* molecules (to be interpreted as μMol) IPTG and ATC. The concentrations of IPTG and ATC decay exponentially in each experiment. Each set of inputs was fed in turn to the 125 circuits. Each simulation was performed until a steady state was reached.

5.1.1 Analysis: Time-Frequency

The motivating example is taken from the work of Guet et al. We analyze the ODE behavior using the non-linear projection discussed in Section 4.1. The results are 125 projection points in the range $[1.3905 \times 10^{-2}, 2.6561 \times 10^{-2}]$ which are divided into 4 classes with our multi-resolution-adaptive binning algorithm. The 4 classes are presented below as images obtained from NYU BioWave. Of significant interest is that the 4 classes are associated to at least as many hypothesized circuit topologies. There is consistency in the classes both in qualitative description of the element functions as well as the derived circuit topology, thus we believe that to a certain extent the low-dimensional clustering of the 125 function encodes the underlying circuitry.

5.1.2 Analysis: Temporal Logic

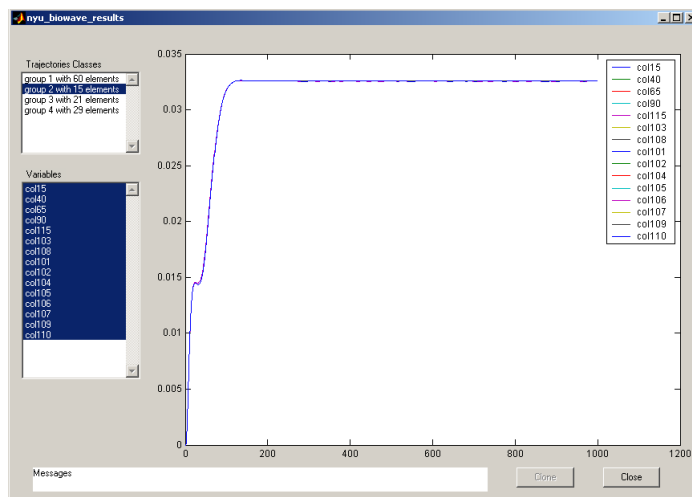
As a simple test of our Simpathica/XSSYS system, we ran a non-traditional analysis of the four sets of trajectories using Simpathica Temporal Logic analysis tool: XSSYS. Simpathica/XSSYS sorted the circuits according to the following properties.

- Circuits exhibiting *switch*-like properties.
- Circuits exhibiting a *boolean* behavior (i.e. showing a combinatorial function of the inputs).

We modified our tool to handle all these cases and proceeded in the following way.

1. Find good *candidate* circuits; call this set C .
These are the circuits that present a variation in outputs given different inputs³.

³This was not really necessary with respect to step 2, as the circuits eliminated would have been classified as the as either boolean constant true, or false.)




Circuit	Comment
	Circuit 104 ($P\lambda_+$, PL1, $P\lambda_-$)

Figure 7: The shape of the trajectories in Group 2 is determined by the topological arrangement of the plasmids in which λ *ci* (C) activates the transcription of one of the other genes, while this gene represses the transcription of λ *ci*. The sample diagram (Circuit 104) reflects this feature. The triple of promoters denotes the structure of the circuit.

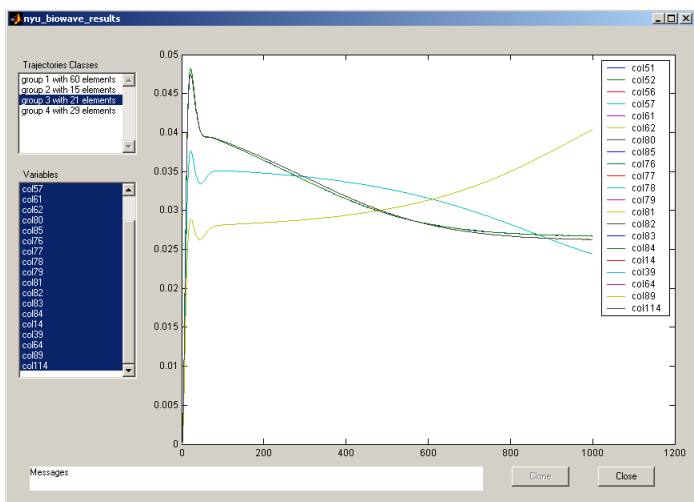
2. Find which circuit $c \in C$ implements one of the basic 2-inputs boolean functions⁴.
3. Find which circuit admits more than 2 output values.

To test for the first property we used the following method. Each circuit was simulated given one of the input pairs in Table 1. The result is a quadruple of traces for each circuit. Next we ran a simple script testing whether the steady state value of each member of the quadruple was above or below a threshold. This corresponded to formulating the following TL query on each element of the quadruple.

`eventually(always(c < threshold)).`

`threshold` was varied in the range $[0.5 \dots 5.0]$ with 0.1 increments. Any circuit c which failed the query for some element of the quadruple was marked as “*potential circuit*.”

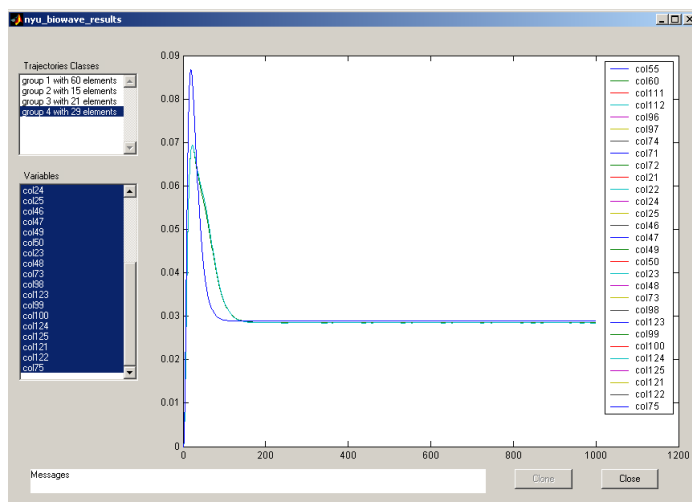
⁴Given two inputs i_1 and i_2 there are 16 possible boolean functions: 0, 1, i_1 , i_2 , $\neg i_1$, $\neg i_2$, OR, AND, NOR, NAND, XOR, NXOR, IF_1_2, IF_2_1, NIF_1_2, NIF_2_1.



Circuit	Comment
	Circuit 76 (Pλ+, PL1, PL1)

Figure 8: The shape of the trajectories in Group 3 is determined by the topological arrangement of the plasmids in which λci represses the transcription of one of the other genes, while this gene represses the transcription of λci . The sample diagram (Circuit 76) reflects this feature. The triple of promoters denotes the structure of the circuit.

The next step was to test which of the potential circuits actually represented a boolean one. This step immediately posed a problem, as certain circuits exhibit a two-valued response to the inputs from Table 1, while other exhibit three-valued response. Moreover, the choice of what constitutes a *high* and *low* response appeared rather arbitrary. To cope with this problem we devised a procedure that automatically constructs TL formulæ of the form



Circuit	Comment
	Circuit 71 (PT, Pλ+, PL1)

Figure 9: Group 4 includes the trajectories whose shape is dominated by the topological arrangement of the plasmids in which λci (C) activates its own transcription and neither *lac* (A) nor *tet* (B) have an affect on the transcription of λci . This feature clearly eliminates the significance of the topological arrangement of the promoters before LAC and TETR. The sample diagram of this group shows lambda CI activating its own transcription, while the relationship is arbitrary, as long as they do not affect λci . Circuit 71 is a sample of the diagrams representing these functions. The triple of promoters denotes the structure of the circuit.

```

eventually(IPTG = 0 and aTc = 0
==> eventually(always(low(c))))
and eventually(IPTG = 0 and aTc = 3
==> eventually(always(high(c))))
and eventually(IPTG = 3 and aTc = 0
==> eventually(always(high(c))))
and eventually(IPTG = 3 and aTc = 3
==> eventually(always(high(c)))).

```

The formula checks whether circuit c represents an OR gate⁵. Mixing the *low* and *high* functions yields tests for all the other 15 two inputs boolean functions. The *low* and *high* functions yields depend on a threshold which can be changed.

⁵The outer *eventually* operator is introduced mostly as a technicality

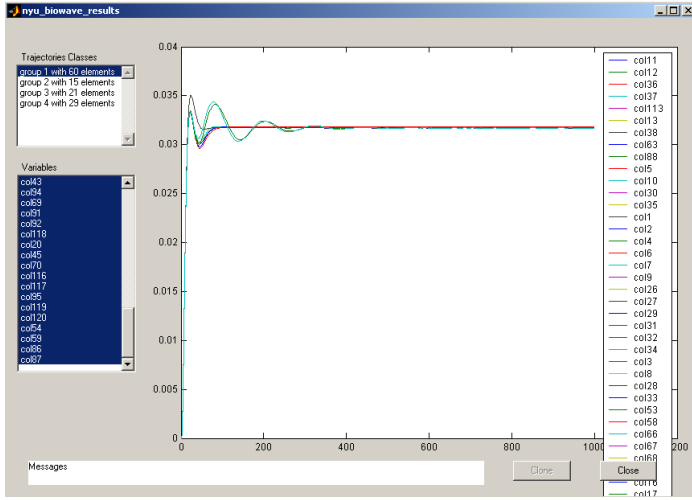


Figure 10: Group 1 incorporates all remaining plasmids. Their topology involves unilateral repression of λci by one of the other genes or by itself. The three sample diagrams reflect these features (the middle example is an oscillator).

Boolean Function	Circuit
\neg IPTG	51 52 56 57 76 77 78 79 80 81 82 83 85
ATC	14 39 64 89 114
ATC \rightarrow IPTG	61 62

Table 2: The classification of potential boolean circuits given a threshold of $1.3 \mu\text{Mol}$. Each number denotes one of the circuits described in [GEHL02].

Table 2 shows which circuits have been identified as which boolean circuit, given a threshold of $1.3 \mu\text{Mol}$.

6 DISCUSSION

In this paper, we have described a set of tools within Simpathica, specifically designed to perform time-frequency analysis of the trajectories of bio-chemical pathways and to classify them into groups for further characterization. Two of the new tools NYU BioSim and NYU BioWave facilitate a user to automate this analysis process and handle a large number of trajectories, obtained either through *in silico* simulation or through *in vitro* or *in vivo* experiments. The capabilities of these systems are illustrated through a detailed analysis of a combinatorial approach to bio-circuit design, following the scheme suggested by Guet et al. [GEHL02].

Circuit	Function	Comment
	\neg IPTG	Circuit 85 $\langle P\lambda_-, PL2, P\lambda_+ \rangle$
	ATC	Circuit 114 $\langle P\lambda_+, PT, P\lambda_- \rangle$
	ATC \Rightarrow IPTG	Circuit 61 $\langle PT, PT, PL1 \rangle$

Table 3: Some of the circuits implementing the logic-combinatorial circuits found with threshold parameter equal to $1.3 \mu\text{Mol}$. Again the triple of promoters denotes the structure of the circuit.

Arguably, much research remains to be done before biological circuit design can be fully and faithfully carried out in this manner, but this style of analysis may ultimately provide a better scheme over other competing approaches based on tedious hand design or *in vitro* evolution. Furthermore, these ideas suggest that our approach will also allow one to study phenotypical properties of a genetic network in wild type, by concomitantly studying a family of mutants and double-mutants obtained by combinatorial knock-outs. Same approach also suggests that the functional properties of a novel gene can be studied by combinatorially mixing it with a family of artificial genetic networks that have already been characterized. Thus, such combination of biological experiments with computational and mathematical tools promises to open up new and exciting opportunities.

References

- [APP⁺03] M. Antoniotti, F. C. Park, A. Policriti, N. Ugel, and B. Mishra. Foundations of a Query and Simulation System for the Modeling of Biochemical and Biological Processes. In *Proc. of the Pacific Symposium of Biocomputing (PSB'03)*, 2003.
- [CW92] R. R. Coifman and M. V. Wickerhauser. Entropy-based Algorithms for Best Basis Selection. *I.E.E.E. Transactions on Information Theory*, 38(2), 1992.
- [EL00] M. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.
- [GEHL02] C. C. Guet, M. B. Elowitz, W. Hsing, and S. Leibler. Combinatorial synthesis of Genetic Networks. *Science*, 296(5572):1466–1470, 2002.

- [KS98] J. Keener and J. Sneyd. *Mathematical Physiology*. Springer-Verlag, 1998.
- [Liò03] P. Liò. Wavelets in bioinformatics and computational biology: state of the art and perspectives. *Bioinformatics*, 19(1):2–9, 2003.
- [Mal99] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1999.
- [Mis02] B. Mishra. A symbolic approach to modeling cellular behavior. In *Proceedings of HiPC 2002, Bangalore, INDIA, December 2002*.
- [Voi91] E. O. Voit. *Canonical Nonlinear Modeling, S-system Approach to Understanding Complexity*. Van Nostrand Reinhold, New York, 1991.
- [Voi00] E. O. Voit. *Computational Analysis of Biochemical Systems A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, 2000.

NYU BioWave is a set of Matlab routines that can be downloaded as a standalone package. NYU BioWave can access NYU BioSim to read data to be analyzed and clustered.

Again, all our software will eventually implement all the interfaces agreed upon by the participants in the DARPA BioSpice working groups.

A Appendix

A.1 Web Resources for Simpathica, NYU BioSim and NYU BioWave

All the software described in this paper is available as part the DARPA BioSpice distribution (see www.biospice.org). The DARPA BioSpice project currently makes releases of the software distribution every six months. Our web site, bioinformatics.nyu.edu may contain more up to date versions of the NYU BioSim, NYU BioWave, Simpathica and other software.

Simpathica is actually a collection of tools: a *pathway editor*, a *pathway simulator*, and an *analysis tool* based on a Temporal Logic model checker. This last module is also known as Simpathica/XSSYS. An OAA (cfr. www.ai.sri.com/oaa) agent providing access to XSSYS is also available on our site.

NYU BioSim⁶ is the core infrastructure of our architecture, as all our tools eventually store their time-indexed data into it. NYU BioSim provides a simple way to import time series data in a variety of formats. Given an application that produces time-series data (e.g. BioCharon from University of Pennsylvania – also available from the DARPA BioSpice distribution), the results can be dumped in NYU BioSim and made available for a number of analysis tools (e.g. NYU BioWave). As an extension that will make NYU BioSim more interoperable with other BioSpice components, we will deploy an OAA agent supporting reading and writing operations on the database.

⁶NYU BioSim is a derivative of NYUMAD, our MGED-compliant Microarray Database and Microarray Analysis tool.