Modeling of the *Saccharomyces cerevisiae* FLR1 Regulatory Network using an S-System Formalism

Dulce Calçada, Susana Vinga, Arlindo Oliveira
KDBIO/INESC-ID, Lisboa, Portugal

dulce.calcada@ist.utl.pt, svinga@kdbio.inesc-id.pt, aml@inesc-id.pt

***Abstract*—This work tackled the problem of finding a mathematical model for the genetic network regulating the stress response of the yeast *Saccharomyces cerevisiae* to the fungicide mancozeb This regulatory network comprises five genes, and an S-system formalism was used to model their interactions. Parameter estimation was accomplished by decoupling the resulting system of nonlinear ordinary differential equations into a larger nonlinear algebraic system, and using the Levenberg-Marquardt algorithm to fit the model's predictions to available experimental data. The introduction of constraints in the parameter estimation procedure, related to the partially known connectivity of the network, was explored. The success of the results obtained was limited, mainly due to the insufficient number of experimental points available, as well as to their poor dynamical variability.**

***Index Terms*—Differential Equations, Gene Regulatory Network, Parameter Estimation, S-system Formalism, *Saccharomyces cerevisiae*.**

I. Background and Problem Definition

THE pattern of expression of a given gene in an organism often depends on a set of other genes, which interact among themselves, forming what is called a genetic regulatory network. Because these networks of interactions are usually large and intricate, and their resulting dynamics very complex and nonlinear, much effort has been put on finding systematic mathematical tools for their modeling and simulation [1]. One such tool is the S-system formalism, which is a variant of the formalism proposed by Biochemical Systems Theory (BST) [2;3]. The S-system formalism represents the dynamics of the biological network as a system of nonlinear ordinary differential equations, describing each component's temporal rate of change (the derivative with respect to time of *e.g.* its concentration) as the sum of one positive and one negative term, accounting, respectively, for its rates of production and degradation:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{M} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{M} X_j^{h_{ij}} \quad , \quad i = 1,2,...,M \qquad (1)$$

In (1), M is the number of components of the network (or the number of state variables of the system), α_i and β_i are non-negative rate constants, and g_{ij} and h_{ij} are real-valued kinetic orders for the production and degradation terms, respectively. The advantage of using canonical formulations such as this is that they provide a consistent mathematical framework for the representation and analysis of the dynamical behavior of biological systems.

The parameters specified by this type of model (vectors α and β, and matrices g and h) are estimated by fitting the model's predictions to experimental data. In terms of the fitting, two approaches may be followed: either

i. the predictions of the numerically integrated system (1) are fitted to experimental time series of the (concentrations of the) components of the network, or
ii. the predictions of the differential equations in (1) are directly fitted to the estimated experimental slopes of these time series.

Besides bypassing the computationally heavy numerical integration steps, the latter approach has the additional advantage of resulting in the replacement of the differential equations (1) with sets of algebraic equations, whose decoupling allows for the parameters pertaining to each state variable to be estimated separately [2]. A potentially problematic requirement of this method has to do with the slope estimates, whose reliability depends on both the existence of a sufficient number of experimental points and an appropriate smoothing of the experimental data.

The estimation of the parameters of highly parameterized nonlinear models of biological systems often reveals a particularly troubling characteristic of these systems: the fact that a large range of parameter values may result in very similar dynamical behavior (and, specifically, in a similarly good fit of the model to the experimental data). In the case of S-system models, this also means that distinct types of interactions among network components may generate the same overall dynamical behavior (as the activation or inhibition effect that a given component has on another is defined by the signs of the corresponding kinetic orders). The fact that the dynamics of these systems are insensitive to particular directions in the parameter search space constitutes a phenomenon called *sloppiness*, and an approach for analyzing the sloppiness of a biological system is to explore the clusters or clouds of parameters which result in a given dynamical behavior.

This work was finantially supported by Fundação para a Ciência e Tecnologia (FCT), contract PTDC/EIA/71587/2006.

II. Case Study: The FLR1 Regulatory Network

The biological system analyzed in this work was the genetic regulatory network that determines the stress-induced expression of *Saccharomyces cerevisiae* (*S. cerevisiae*) gene FLR1, in response to the fungicide mancozeb. This network comprises five interacting genes (the state variables) and its mathematical formulation was based on the S-system formalism.

The aim was to estimate the 60 resulting parameters (two 5-by-1 α and β vectors, and two 5-by-5 g and h matrices), by fitting the slopes of each gene's relative concentration at each time point (as estimated from the corresponding experimental time series) to the derivatives predicted by the mathematical model developed. The Levenberg-Marquardt algorithm was used to minimize a cost-function defined as the sum of squared residuals between the predicted derivatives and the estimated experimental slopes at each time point, subject to a number of nonlinear constraints [3].

Because of the system's anticipated sloppiness, the approach described by Vilela et al. [3] was used for parameter estimation, with admissible parameter sets being estimated by a Monte Carlo process combined with the optimization algorithm.

III. Pre-Processing of Experimental Data

Five sets of experimental data were available for this work [4]. Each set consists of five time series of the relative concentrations of the transcripts of five *S. cerevisiae* genes (parental strain BY4741) involved in its stress response to the fungicide mancozeb, measured over the course of 11 hours, at 8 non-equally spaced time instants. The genes are FLR1, YAP1, PDR3, RPN4 and YRR1 and the relative concentrations of their transcripts (Flr1p, Yap1p, Pdr3p, Rpn4p and Yrr1p, respectively) were obtained as ratios of the levels of the corresponding mRNA to the levels of ACT1 mRNA in the cells. The ratios obtained for control conditions (absence of mancozeb) were set to 1 and the remaining values were obtained relative to that control. Each data set corresponds to experiments carried out using the wild-type *S. cerevisiae* strain BY4741, labeled *wt*, and four deletion mutants, missing genes YAP1, PDR3, RPN4 and YRR1, and labeled Δyap1, Δpdr3, Δrpn4 and Δyrr1, respectively.

In terms of the FLR1 gene, the yeasts' stress response to mancozeb is characterized by a transient initial activation of FLR1 expression, followed by a decline in FLR1 transcript levels, until full adaptation to mancozeb is attained (e.g. in the *wt* data set). Because of this transient response, the initial period of the experiments was more finely sampled, with six measurements taken in the first 2 hours, and the two final measurements taken at the 6th hour and at the 11th hour.

The authors sought to ascertain the regulatory roles of genes YAP1, PDR3, RPN4 and YRR1 in the control of FLR1 activation, and their analysis resulted in the proposal of a network specifying the interactions among these four transcriptional regulators and the FLR1 gene.

The parameter estimation algorithm used in this work requires the computation of the derivatives of the experimental time series. Because of the limited number of experimental data points and of noise in the corresponding measurements, an accurate estimate for these derivatives requires that the data be smoothed. Thus, cubic smoothing splines [5] were constructed for every experimental time series using function `csaps` of the MATLAB® Spline Toolbox. The smoothing spline f = `csaps(x,y,p)` minimizes

$$p\sum_{j=1}^{n} w(j)\left|y(:,j)-f(x(j))\right|^2+(1-p)\int \lambda(t)\left|D^2 f(t)\right|^2 dt \quad (2)$$

The first term in (2) is called the error measure, and the second term the roughness measure. Parameter p is called the smoothness parameter as it determines the relative weight of the two contradictory demands of having f be smooth and simultaneously be close to the data. For $p=0$, f will be the least-squares straight line fit to the data, while for $p=1$, it will be the data's "natural" cubic spline interpolant. Parameter w in the error measure is a vector of weights, and parameter λ in the roughness measure is a piecewise constant weight function, which allows one to force f to be smoother (by making the weight function larger) or, on the contrary, closer to the data (by making the weight functions smaller), in some parts of the spline interval than in others.

In the present case, x is the vector of measurement instants and y is the vector of transcript levels of a given gene, in a given data set. Parameter w was set to be a vector of ones the same size as x, since early experiments, in which it was set to be the inverse of the measurements variance (computed, when available, from 3 independent measurements), yielded poor results. Parameters p and λ were chosen, for each time series, by trial and error. In the case of parameter p, the values typically used were 0.8 or 0.97 but values as low as 0.3 and 0.5 were found to be more adequate in some instances. As for parameter λ, its value also varied, but, in general, more roughness was allowed at the initial time points, where he transient response occurs (smaller λ for those points).

The splines were evaluated at 67 equally spaced time points, and their derivatives were obtained at the same time points, by using function `fnder`. Fig. 1 below shows the plots of the experimental and splined time series of the five aforementioned genes, in the *wt* data set. Fig. 2 shows the corresponding derivatives. The same data is shown in Fig. 3 and Fig. 4, respectively, for the Δyap1 data set.

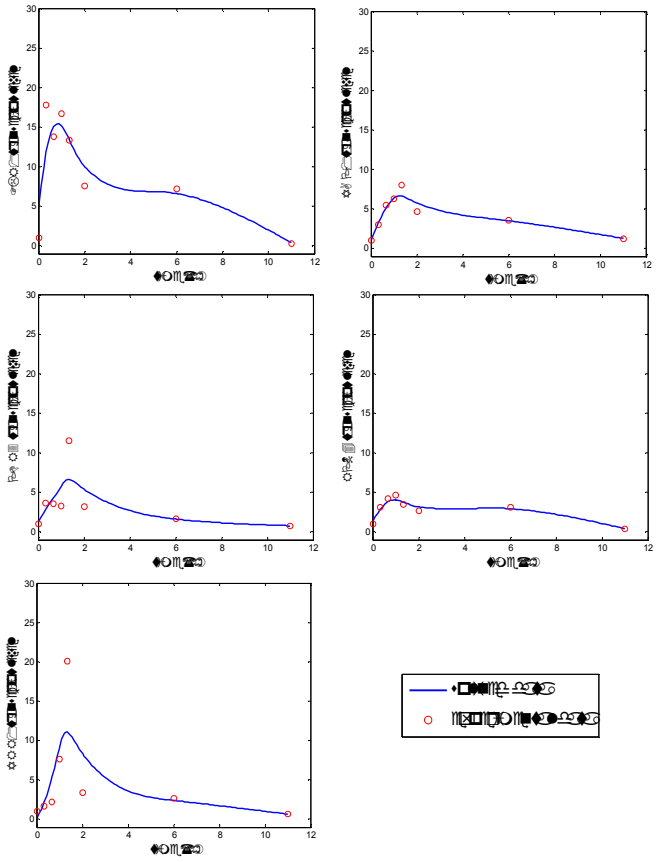

Fig. 1. Experimental time series (red circles) of the transcript levels of FLR1, YAP1, PDR3, RPN4 and YRR1 in the wild-type strain of *S. cerevisiae* (*wt* data set). The blue lines are cubic smoothing splines.

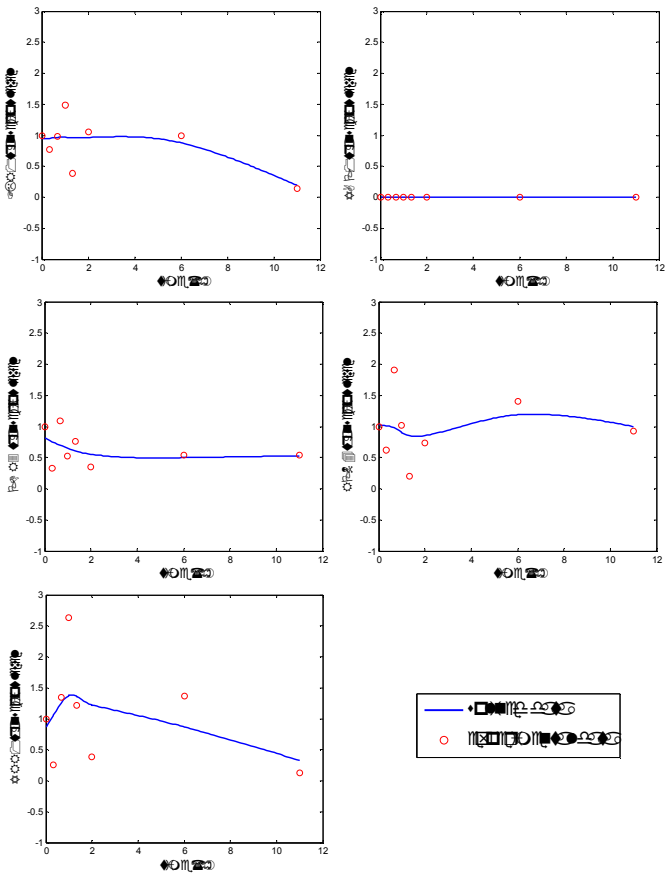

Fig. 3. Experimental time series (red circles) of the transcript levels of FLR1, YAP1, PDR3, RPN4 and YRR1 in the YAP1 deletion mutant of *S. cerevisiae* (Δyap1 data set). The blue lines are cubic smoothing splines.

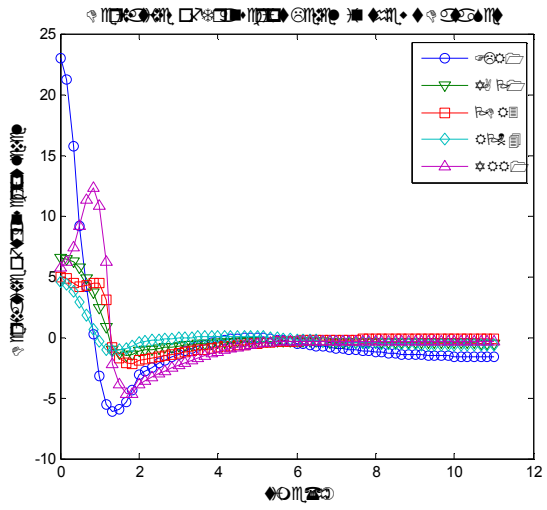

Fig. 2. Time series of the transcript levels derivatives of genes FLR1, YAP1, PDR3, RPN4 and YRR1 in the *wt* data set, as obtained from the splines used to smooth the time series, at 67 equally spaced time points.

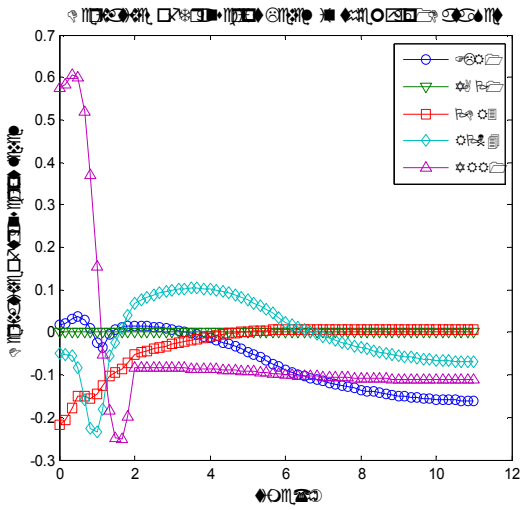

Fig. 4. Time series of the transcript levels derivatives of genes FLR1, YAP1, PDR3, RPN4 and YRR1 in the Δyap1 data set, as obtained from the splines used to smooth the time series, at 67 equally spaced time points.

Although the results were satisfactory for the *wt* data set (Fig. 1), the same cannot be said about the Δyap1 data set (Fig. 3). Indeed, the amount of noise seemed to be greater in the latter, and, had more points been sampled, data smoothing would probably have yielded a constant function for every time series. This problem also arose in a considerable portion of the time series in the other three data sets, leading to the conclusion that the few available measurements were not informative enough to build reliable splines.

As an alternative to cubic smoothing splines, the automated smoother described in [6] was also applied to the data, but, despite its handling of nonstationary noise structures, the results were not satisfactory. The limited success of this smoothing method may have been due to the limited number of experimental points available and, perhaps more importantly, to the fact that the experimental points were not equally spaced.

IV. Modeling and Parameter Estimation

As stated earlier, the mathematical modeling of the FLR1 regulatory network was based on the S-System formalism, with the state variables being the transcript levels of its five interacting genes: X_1=FLR1, X_2=YAP1, X_3=PDR3, X_4=RPN4, and X_5=YRR1.

Parameter estimation was based on the decoupling of the resulting system of ordinary differential equations, and the sets of admissible parameters were estimated using the method described in [3]. The computational implementation used was that made freely available by the authors in the form of a set of MATLAB® scripts. The main script is `EO_mainf` and its syntax is

```
result = EO_mainf(X,S,p)
```

with X being the experimental time series, S its corresponding estimated slopes, and p a structure of optional parameters provided to the algorithm. X and S are thus matrices with a number of rows equal to the number of (splined) time points, and a number of columns equal to the number of network components.

Eight distinct settings were considered for parameter estimation, varying both in the data sets used, and in the topological constraints applied to the network. For each of these eight settings, the algorithm was run 100 times, with differing initial conditions (as explained at the end of this section). In terms of the data used, two alternatives were explored:

i. using all five data sets, resulting in 335-by-5 X and S matrices (67×5=335 splined data points for each gene);
ii. using only the *wt* data set, resulting in 67-by-5 X and S matrices (67 splined data points for each gene).

As for the constraints applied to the network, four settings were tested, by imposing zero-valued entries in the kinetic orders matrices g and h. In practice, this was done by inputting the fields $p.G$ and $p.H$ of structure p as matrices of ones and zeros, according to whether or not a given gene was allowed to affect another gene's production and degradation rate. The constraints imposed were based on the (positive or activation-type) interactions specified in the putative network proposed in [4], of which a simplified version is reproduced in Fig. 5. Because there was no information regarding negative interactions among the genes, the constraints imposed on $p.H$ simply reflect the fact that the rate of degradation of each gene should depend solely on its own concentration level and not on any other component's.

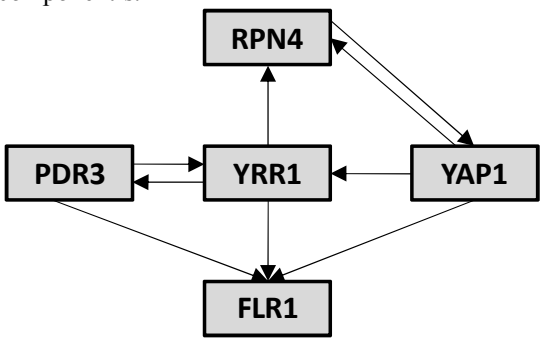

Fig. 5. Putative FLR1 regulatory network proposed in [4]. The black arrows represent the activation effect that a given gene has on the level of expression of another (up-regulation only).

The four tested settings were:

i. a fully unconstrained network, where each gene is allowed to have an effect on every other gene's production and degradation terms, i.e. for which

$$p.G = p.H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3)$$

ii. a fully constrained network, where the effect of a given gene on another gene's production term was determined by the putative network proposed in [4] (with an additional dependence of the production rate of a gene on its own expression level), and its degradation term was made to depend only on its own level of expression, i.e. for which

$$p.G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad (4)$$

$$p.H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

iii. a network with constraints on the production term but not on the degradation term, i.e. for which $p.G$ is as defined in (4) and $p.H$ is as defined in (3);
iv. a network with constraints on the degradation term but

not on the production term, i.e. for which *p.G* is as defined in (3) and *p.H* is as defined in (5).

The remaining fields of structure *p* for which the default values were not used were *p.iter*, *p.ubB* and *p.int*. Parameter *p.iter* (which is the number of iterations for the optimization algorithm and whose default value is 300) was set to 500; parameter *p.ubB* (which is the upper boundary value of the constant rates β and whose default value is 300) was set to 100; and parameter *p.int* (which is a positive scalar used to calculate the initial values for the optimization algorithm and whose default value is 1) was made to vary between different runs of the algorithm – if *k* was the run index, then *p.int* was set to *k/5*.

V. Results

Two criteria were defined for evaluating the performance of the parameter estimation algorithm in each of the 8 settings tested. The first criterion, C_1, was the lowest cost function value achieved for each setting in the 100 runs of the algorithm, and the second criterion, C_2, was the percentage of runs reaching a cost function value below a certain threshold (which was set to 1000). The results of this evaluation are shown in Table I.

TABLE I
CRITERIA C_1 AND C_2 FOR EACH SETTING

Setting	Data Sets used	Constraints	C_1	C_2
1	all	none	-	0
2	all	on *g* and *h*	-	0
3	all	only on *g*	-	0
4	all	only on *h*	-	0
5	*wt*	none	108.82	60
6	*wt*	on *g* and *h*	945.92	100
7	*wt*	only on *g*	141.60	100
8	*wt*	only on *h*	310.71	99

According to these criteria, the worst overall settings were those for which all five data sets were used to estimate the parameters. This may be explained by the fact the poor dynamic variability exhibited by the slopes of a large portion of the time series in the non-wild-type data sets. Specifically, in each of the four sets referring to deletion mutants, Δyap1, Δpdr3, Δrpn4 and Δyrr1, the time series of the corresponding gene is constant. As the authors of the method point out [3], this may lead the algorithm to produce spurious results, due to numerical problems with the inversion of an ill-conditioned matrix. Within settings 5 to 8 (for which only the *wt* data set was used for parameter estimation) the best results were achieved when no constraints were imposed on the topology of the network (setting 5), while the worst results were obtained for the fully constrained topology.

Further analysis of the optimization results focused on the agreement of the predictions of the integrated version of (1) with the experimental data. Thus, defining the error (in the model) as the sum of squared residuals between the predictions of the integrated system (1) and the experimental data, two additional criteria for the evaluation of the results were devised: criterion C_3, corresponding to the lowest error achieved for each setting in the 100 runs of the algorithm; and criterion C_4, corresponding to the percentage of runs achieving an error below a certain threshold (which was set to 5000).

The predictions of the 100 models in each setting were obtained for the five time series in each data set, via numerical integration of the system, with the five corresponding initial conditions in each data set. The numerical integration of the system was performed with the simple Euler method, since the stiff solvers available in `MATLAB®` (`ode15s` and `ode23s`) could not successfully handle it. Time steps of 1/60 h were used, and a safeguard for the prediction of negative values (as well as numerical errors resulting in infinite and `NaN` values) was applied: if such values were predicted, they were changed into the value at the previous time step. Table II below shows the value of evaluation criterion C_3 for each setting and data set.

TABLE II
CRITERION C_3 FOR EACH SETTING AND DATA SET

Setting	Data Set				
	wt	Δyap1	Δpdr3	Δrpn4	Δyrr1
1	1270.7	18.189	691.24	832.84	102.69
2	14643	18.189	691.24	832.84	102.69
3	Inf	18.189	7×10^{151}	832.84	102.69
4	1450.2	18.189	691.24	832.84	102.69
5	727.36	18.189	691.24	832.84	102.69
6	11050	18.189	691.24	832.84	102.69
7	1018.3	18.189	691.24	832.84	102.69
8	215.70	18.189	691.24	832.84	102.69

Because these are absolute values, one cannot compare the errors among data sets (that is, the columns of Table II are not comparable). Nonetheless, a striking result is that the same lowest error was obtained for nearly all the settings in each data set referring to a deletion mutant (the only exception was the Δyap1 data set, for which the lowest error achieved in setting 3 was exceedingly large). In fact, the best prediction in settings 1 to 8 was equivalent for every one of these deletion mutant data sets, yielding, for each gene, transcript levels that were constant in time and equal to their corresponding initial level (see in Fig. 8 the predictions of the models in setting 8 for the Δyap1 data set).

Regarding the predictions for the *wt* data set, poor results were once again obtained for the settings imposing a fully constrained network topology (settings 2 and 6), but it was setting 3, referring to a production-term-constrained network whose parameters were estimated using all data sets, that yielded the worst results (with an "infinite" value being returned for the error). The best results were obtained for settings 8 and 5, referring, respectively, to a degradation-term-constrained network (constrained matrix *h*) and a totally unconstrained one.

Table III shows the value of evaluation criterion C_4 (percentage of runs reaching an error below 5000), for each setting and data set.

TABLE III

CRITERION C_4 FOR EACH SETTING AND DATA SET. IN PARENTHESIS IS THE PERCENTAGE OF RUNS SATISFYING BOTH C_2 AND C_4.

Setting	Data Set				
	wt	Δyap1	Δpdr3	Δrpn4	Δyrr1
1	100 (0)	100 (0)	100 (0)	100 (0)	100 (0)
2	0 (0)	100 (0)	100 (0)	100 (0)	100 (0)
3	0 (0)	100 (0)	0 (0)	100 (0)	100 (0)
4	68 (0)	100 (0)	100 (0)	100 (0)	100 (0)
5	71 (51)	100 (60)	100 (60)	100 (60)	100 (60)
6	0 (0)	100 (100)	100 (100)	100 (100)	100 (100)
7	20 (20)	100 (100)	100 (100)	100 (100)	100 (100)
8	64 (64)	100 (99)	100 (99)	100 (99)	100 (99)

Although a great percentage of runs pertaining to settings 1-4 satisfy criterion C_4, none satisfy both C_2 (referring to the error in the derivatives) and C_4. On the contrary, in settings 5 to 8, there is always at least one run satisfying both criteria, with the exception of setting 6 (fully constrained network), when trying to predict the values of the *wt* data set. The graphical depictions of some of the predictions obtained for these models are analyzed next. Fig. 6 below shows the time courses predicted for the *wt* data set by the models satisfying C_2 and C_4 in setting 5.

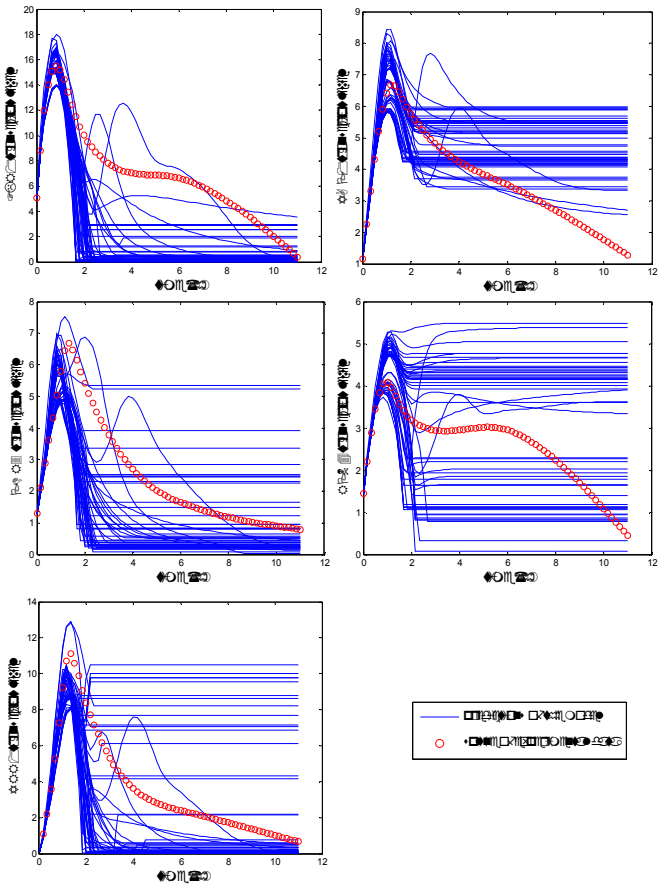

Fig. 6. Predicted time courses of the best runs from setting 5 for the initial conditions in the *wt* data set.

The time courses in Fig. 6 show that several parameter sets resulted in atypically constant predictions in the final period of the experiment. Although it was at first suspected that this was simply the result of the safeguards introduced in the numerical integration routine, it was found that the same predictions arose when no safeguards were used. An alternative explanation is that due to the larger number of parameters that had to be estimated for this fully unconstrained network, as well as to the fact that some of the time series are approximately collinear (*e.g.* PDR3 and YRR1) and have constant slopes at the end of the experiment, the algorithm's convergence was misled and a sub-optimum parameter set was obtained.

Fig. 7 shows the time courses predicted for the *wt* data set by the models satisfying C_2 and C_4 in setting 8.

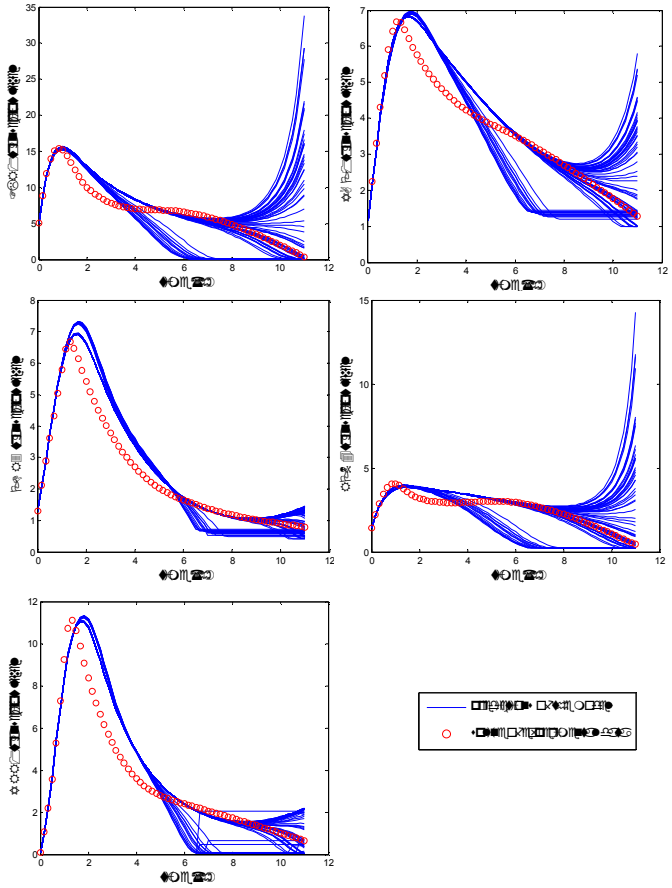

Fig. 7. Predicted time courses of the best runs from setting 8 for the initial conditions in the *wt* data set.

For this setting, the parameter sets satisfying C_2 and C_4 do not result in time courses with constant final "tails" higher than the experimental ones (as was the case with setting 5), but instead, one of two distinct dynamic behavior clusters obtained does show a very quick rise at the final time instants, for nearly every gene. Also, the time courses predicted for each gene are, in general, more consistent than those obtained by the models in setting 5.

Fig. 8 shows the time courses predicted for the Δyap1 data set by the models satisfying C_4 in setting 8. These results are very different from the ones on the two previous figures, since there is absolutely no variability in the models' predictions, i.e. all 100 models predict, for each gene, the same time course. Though at first sight surprising, this may be partially explained by the low initial transcript levels observed for all the genes.

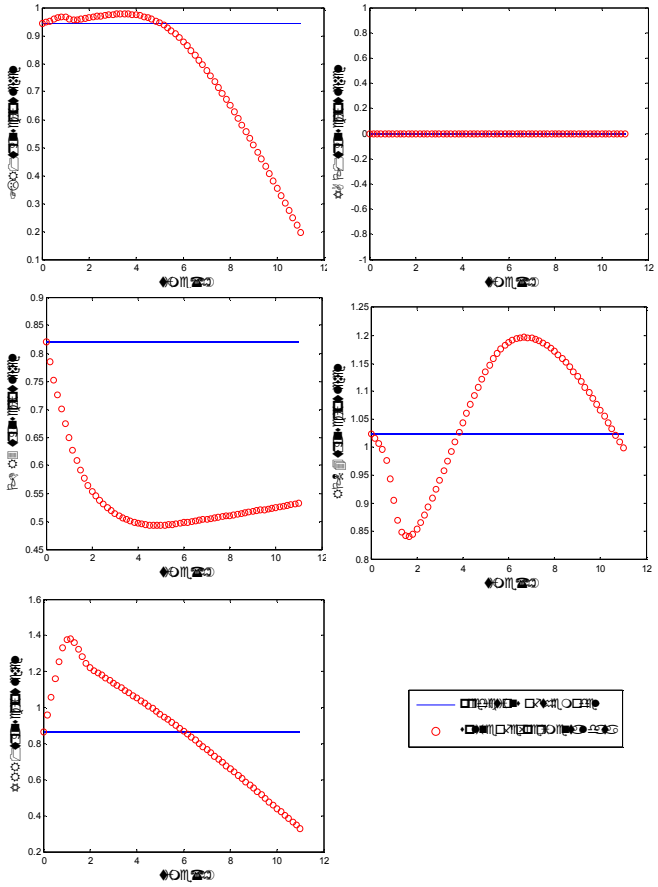

Fig. 8. Predicted time courses of the best runs from setting 8 for the initial conditions in the Δyap1 data set.

Given the difficulties just discussed, it would not be useful to embark on a more detailed analysis of the several parameter sets obtained for each setting, as an analysis of the system's sloppiness requires that a sufficient number of parameter sets resulting in *similar dynamical behavior* be obtained.

VI. Conclusions

The main limitations in the results of this work are related to particularities of the experimental data. Firstly, too small a number of experimental points were available, resulting in the limited reliability of the splines constructed to smooth the data. This had, in turn, a negative effect on the estimation algorithm, since it crucially depends on the time series slopes estimated from the experimental data.

On the other hand, because the experiments dealt with mutant *S. cerevisiae* strains, each lacking a certain gene, the time course for that gene's transcript level was always zero. Moreover, when the presence of this gene is required for the expression of other genes in the network (*e.g.* YAP1 is required for FLR1 expression), this means that the time course for those genes will also be zero throughout the whole experiment. The resulting poor dynamic range of the estimated slopes, as well as the presence of collinear time series, pose serious difficulties to the parameter estimation algorithm, and may mislead its convergence.

Further modeling efforts, in the context of the continuous-time S-system formalism, should therefore depend on the obtaining of a larger and more diverse set of experimental data. Alternatively, discrete modeling of the network could perhaps be attempted with the present data [7;8], but the results should prove more qualitative than quantitative.

VII. Acknowledgements

This work was supported by projects *PTDC/BIO/72063/2006, Characterization and modeling of a specific transcriptional regulatory network required for multidrug resistance in yeast* and *PTDC/EIA/71587/2006, Dyablo: Models for the Dynamic Behavior of Biological Networks*, financed by Fundação para a Ciência e Tecnologia.

References

Bibliography

[1] H. De Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67-103, 2002.

[2] M. Vilela, I. C. Chou, S. Vinga, A. Vasconcelos, E. Voit, and J. Almeida, "Parameter optimization in S-system models," *BMC Systems Biology*, vol. 2, no. 1, p. 35, 2008.

[3] M. Vilela, S. Vinga, M. Maia, E. Voit, and J. Almeida, "Identification of neutral biochemical network models from time series data," *BMC Systems Biology*, vol. 3, no. 1, p. 47, 2009.

[4] M. C. Teixeira, P. J. Dias, T. Simões, and I. Sá-Correia, "Yeast adaptation to mancozeb involves the up-regulation of FLR1 under the coordinate control of Yap1, Rpn4, Pdr3, and Yrr1," *Biochemical and Biophysical Research Communications*, vol. 367, no. 2, pp. 249-255, Mar.2008.

[5] C. de Boor, *A Practical Guide to Splines* Springer Verlag, 2001.

[6] M. Vilela, C. Borges, S. Vinga, A. Vasconcelos, H. Santos, E. Voit, and J. Almeida, "Automated smoother for the numerical decoupling of dynamics models," *BMC Bioinformatics*, vol. 8, no. 1, p. 305, 2007.

[7] A. Rubinstein, V. Gurevich, Z. Kasulin-Boneh, L. Pnueli, Y. Kassir, and R. Y. Pinter, "Faithful modeling of transient expression and its application to elucidating negative feedback regulation," *Proceedings of the National Academy of Sciences*, vol. 104, no. 15, pp. 6241-6246, Apr.2007.

[8] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, "The yeast cell-cycle network is robustly designed," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 14, pp. 4781-4786, Apr.2004.