Algorithmic Parameter Space Reduction of a Systems Biology Model: A Case Study

Sin, Celine

UNIVERSITY OF CALIFORNIA

Los Angeles

Algorithmic Parameter Space Reduction of a Systems Biology Model:

A Case Study

A thesis submitted in partial satisfaction

of the requirements for the degree Master of Science

in Biomedical Engineering

by

Celine Sin

2012

ABSTRACT OF THE THESIS

Algorithmic Parameter Space Reduction of a Systems Biology Model: A Case Study

by

Celine Sin

Master of Science in Biomedical Engineering

University of California, Los Angeles, 2012

Professor Joseph DiStefano III, Chair

Ordinary differential equation (ODE) models are often used to quantitatively describe and predict the dynamic responses of biological and other systems. Models with many parameters, limited measurement data and in need of quantification are typically unidentifiable from available input/output data. Even models that are structurally identifiable can be difficult to quantify in practice from limited data. For overparameterized models (OPMs), it is often helpful to simplify the model, by rationally reducing the dimensionality of the parameter space. This is done by finding a set of "key parameters" to estimate, a subset that best represents the dominant model dynamic responses. OPMs are often characterized by pairwise parameter correlations close to 1 in magnitude and at least some unacceptably large parameter estimation variances. The goal is to get the best fit possible with a smaller number of parameters, each with acceptable variances. Several published methods for selecting the key parameter subset are based on parameter sensitivity analysis and/or analysis of the parameter covariance matrix estimated from the input/output data.

We apply a combination of these methods to an overparameterized candidate model of tumor suppressor protein p53. The model comprises of 4 ODEs, 23 unknown parameters, and noisy output measurements of the 4 state variables and the input. Three least sensitive and highly correlated parameters were isolated from the analysis and fixed to nominal values. This reduced the parameter search space and yielded substantially improved numerical identifiability properties for the resulting simplified model which fitted the data equally well, using both global and local search algorithms.

The thesis of Celine Sin is approved.

Elliot Landaw

Matteo Pellegrini

Joseph DiStefano III, Committee Chair

University of California, Los Angeles

2012

To Ostrich,

Broggy,

and Mr. Food.

TABLE OF CONTENTS

Acknowledgements

What a great journey so far! I am so grateful for all the wonderful people in my life – this thesis could not have been completed without them.

Many thanks:

...To my advisor Professor Joe DiStefano for many things. It has been a (very) long road full of administrative struggles, and he has continually guided me and supported me. Working together has been a pleasure and honor and I will fondly remember our experiments in the kitchen and in the lab.

...To members of my thesis committee Professor Elliot Landaw and Matteo Pellegrini, for their effort to serve, and their patience.

...To Professor Carol Newton, for two wonderful quarters in her courses, exploring and learning.

...To the support teams and forum contributors of COPASI, AMIGO and MATLAB.

...To Family, to friends. Especially those who have inadvertently lengthened this journey.

...And Lastly, to my constant(ly sleepy) companion and teammate, double-dash driver, partner in (cr/t)ime, sous chef, feather supplier, portable home carrier, babysitter, bedtime story teller, MoOW delivery boy and best friend.

I. Introduction

Mathematical models are quantitative representations of systems, using mathematical concepts and symbolic language. Indispensible for systems biology, models can be used to better understand biological systems, study the effects of different stimuli and make predictions about their behavior. Examples of fairly large systems biology models include: the JAK-STAT pathway (Quaiser, Dittrich et al. 2011), the IL-6 pathway (Daun, Rubin et al. 2008; Chu and Hahn 2009; Huang, Chu et al. 2010), the NF-κβ gene network (Yue, Brown et al. 2006), and the pentose phosphate pathway (Degenring, Froemel et al. 2004).

Mathematical models are developed by translating physical phenomena into mathematical equations. As details are uncovered about underlying physical phenomena (e.g. pathways, molecular interactions, rates, etc.), models to represent the system becomes more complex as the new information is integrated (Schmidt, Madsen et al. 2008; Chu and Hahn 2009; Quaiser, Dittrich et al. 2011). When enough quantitative information becomes available, models can be simulated and analyzed. However, complexity, can be quite troublesome for the quantification process. Increases in model complexity usually involve even larger numbers of model parameters, which cannot be reliably estimated from available data (Chu and Hahn 2009).

Parameters usually cannot be measured directly. They must be estimated from measured input-output data. This may not be possible or practical, for two reasons. The model may not be *structurally identifiable (SI)*, meaning some parameters cannot be solved for, even from perfect, unlimited, ideal data. Even if the model IS structurally identifiable, it may not be *numerically identifiable (NI)*, meaning parameter estimates from limited and noisy data might have too much variability to be useful (Distefano and Cobelli 1980). Identifiability (or lack there of) often can be improved by simplifying the model, either through a reduction of state variable dimensionality or number of parameters to be estimated. In this thesis, we address overparametized models (OPMs) rationally by reducing the dimensionality of the parameter space, choosing a subset of "key identifiable parameters" that best represents the dominant model dynamical responses.

OPMs are often characterized by pairwise parameter correlations close to 1 in magnitude and at least some unacceptably large parameter estimation variances. The goal here is to get the best fit possible with a smaller number of parameters, each with acceptable variances. Many published methods for selecting the key parameter subset are based on parameter sensitivity analysis and/or analysis of the parameter covariance matrix estimated from input-output data (Brun, Kuhni et al. 2002; Smets, Bernaerts et al. 2002; Degenring, Froemel et al. 2004; Yue, Brown et al. 2006; Banks, Dediu et al. 2007; Daun, Rubin et al. 2008; De Pauw, Steppe et al. 2008; Chu and Hahn 2009; Cintron-Arias, Banks et al. 2009; Doherty and Hunt 2009; Huang, Chu et al. 2010; Quaiser, Dittrich et al. 2011).

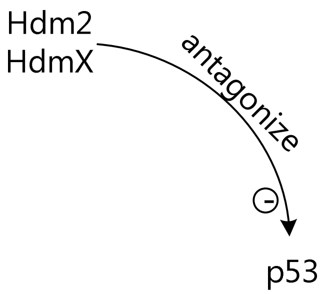
We apply the methods developed by (Daun, Rubin et al. 2008; Chu and Hahn 2009) and a variant on that by (Cintron-Arias, Banks et al. 2009) to an overparametized candidate model of tumor suppressor protein p53 dynamics.

II. Model Description

p53 is an essential protein for tumor suppression. In stressful situations resulting in DNA damage, the role of p53 is to stop cell cycle progression (cell arrest) and start DNA repair. In cases when DNA repair is not possible, p53 initiates apoptosis. Cells without functional p53 give rise to progeny with increasing genetic abnormalities, often becoming cancers – mutations in the p53 gene account for half of human cancers (Wang, Wade et al. 2007)

In normal unstressed conditions, p53 activity in the cell is low, negatively regulated by Hdm2 and HdmX. Hdm2 targets p53 for degradation, and both Hdm2 and HdmX antagonize the transactivation of p53 (Wang, Wade et al. 2007).

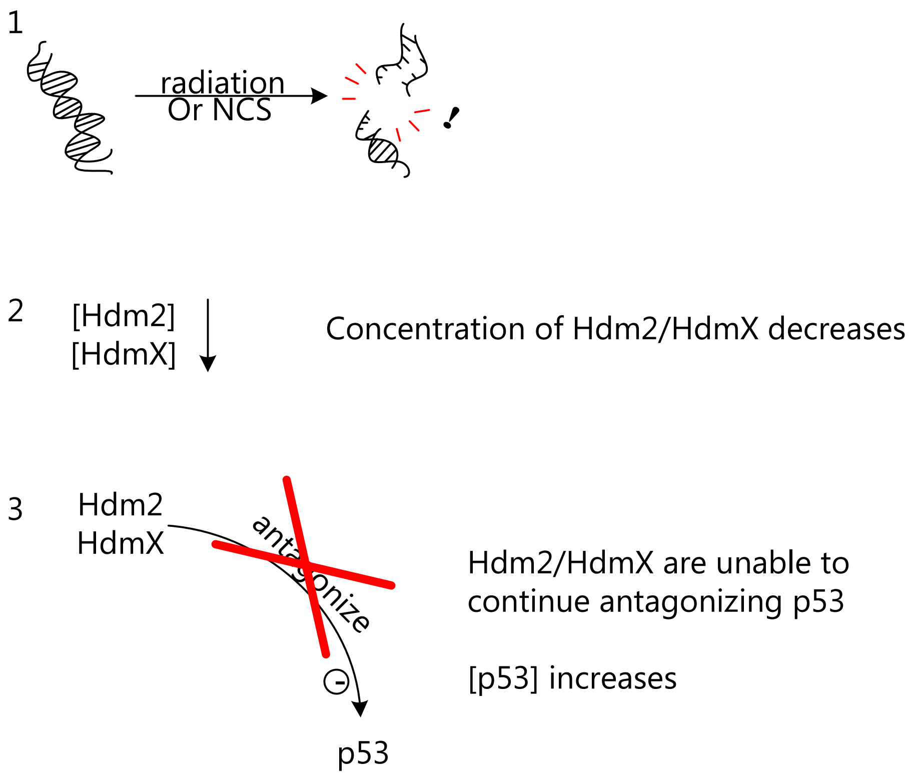
Cell with no DNA damage



Negative regulation
by Hdm2 and HdmX
keeps p53 activity low

During times of stress (e.g. oxidative, osmotic, DNA damage, among others), the stability of Hdm2 and HdmX decreases, reducing the ability of Hdm2 and HdmX to antagonize p53. p53 activity is up-regulated and activated to induce cell repair / arrest / apoptosis (Wang, Wade et al. 2007).

Cell with DNA damage


1
radiation
Or NCS
!
2
[Hdm2]
[HdmX]
Concentration of Hdm2/HdmX decreases
3
Hdm2
HdmX
antagonize
-
p53
Hdm2/HdmX are unable to continue antagonizing p53
[p53] increases

A Candidate Model for p53/Hdm2 dynamics

This paper focuses on analysis and modification of a mathematical ordinary differential equation (ODE) model developed by DiStefano and coworkers (DiStefano unpublished). It describes the protein and mRNA response of tumorigenic mammary epithelial (MCF7) cells following exposure to Neocarzinostatin (NCS), a small molecule-protein complex that causes double stranded DNA breaks. The model is comprised of 4 ODEs and 25 parameters. 23 parameters are unknown.

The ODEs represent four of the species which comprise the p53 pathway: Hdm2 (x_1), HdmX (x_2), p53 (x_3) and Hdm2 mRNA (x_4).

$$\frac{dx_1}{dt} = p_1 x_4(t-\tau) - p_3 x_1 - p_4\left(\frac{x_1^2}{p_5 + x_1}\left(1+\frac{p_6 u}{p_7+u}\right)\right) \tag{1}$$

$$\frac{dx_2}{dt} = p_8 - p_9 x_2 - p_{10}\left(\frac{x_1 x_2}{p_{11} + x_2}\left(1+\frac{p_{12} u}{p_{13}+u}\right)\right) \tag{2}$$

$$\frac{dx_3}{dt} = p_{14} - p_{15} x_3 - p_{16}\frac{x_1 x_3}{p_{17} + x_3}(1 - p_{18} u) \tag{3}$$

$$\frac{dx_4}{dt} = p_{19} - p_{20} x_4 + p_{21}\left(\frac{x_3^n}{p_{22}^n + x_3^n}\right)(1 + p_{23} u)(1 - p_{24} x_1)(1 - p_{25} x_2) \tag{4}$$

where p19 is constrained by the following equation

$$p_{19} = p_{20} - p_{21}\left(\frac{1}{p_{22}^n + 1}\right)(1 - p_{24})(1 - p_{25})$$

NOTE: We set n (the hill coefficient for p53-dependant mRNA transcription) = 4 (DiStefano unpublished). We do not estimate this parameter.

Nominal parameter estimates from our best fit of the data to the full unreduced model are given below. This is not a unique set, because parameter correlations are high for many combinations, with 4-6 clusters of different solutions dominating the results, all fitting the data equally well. This is strong evidence for overparameterization.

Parameter	**Description**	**Value**	**CV (%)**	**dimension**
p01	Hdm2 production rate	0.00910	33	time^{-1}
p02	Hdm2 transcription and transport time delay	0	n/a	time
p03	Hdm2 basal degradation rate	0.00316	114	time^{-1}
p04	Hdm2 degradation rate (by ubiquitination)	0.0506	1050	time^{-1}
p05	Hill K for Hdm2 ubiquitination	85.3	156	concentration
p06	Hill V for Hdm2 ubiquitination	89.7	886	unitless

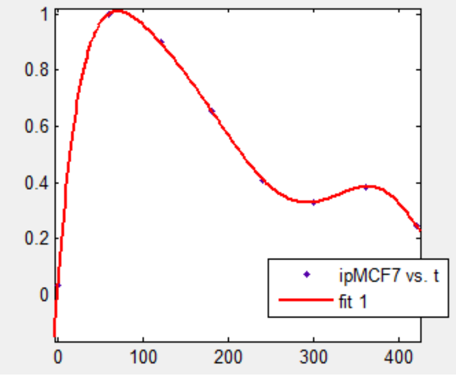
p07	Hill K for Hdm2 phosphorylation by kinase	1.77	270	concentration
p08	HdmX production rate	0.0263	32.7	time^{-1}
p09	HdmX basal degradation rate	0.0244	47.9	time^{-1}
p10	HdmX degradation rate (Hdm2 dependent)	0.0217	272	time^{-1}
p11	Hill K for HdmX ubiquitination	2.91	199	concentration
p12	Hill V for HdmX ubiquitination	4.29	136	unitless
p13	Hill K for HdmX phosphorylation by kinase	0.931	236	concentration
p14	p53 production rate	0.085	3.67	time^{-1}
p15	p53 basal degradation rate	0.00727	23.9	time^{-1}
p16	p53 degradation rate (Hdm2 dependent)	0.0411	14.9	time^{-1}
p17	Hill K for p53 kinase phosphorylation	3.13E-06	1.03E+07	concentration
p18	Inhibition of p53 degradation by kinase phosphorylation	0.315	81.7	unitless
p19	hdm2 mRNA production rate	0.0523	n/a	time^{-1}
p20	hdm2 mRNA basal degradation rate	0.0486	90.9	time^{-1}
p21	hdm2 mRNA transcription rate (p53 dependent)	1.64	61.3	time^{-1}
p22	Hill K of hdm2 mRNA transcription (p53 dependent)	3.24	23.8	unitless
p23	Stimulation of mRNA transcription by p53 phosphorylation	3.68E-06	2.36E+07	unitless
p24	Inhibition of mRNA transcription by Hdm2	0.128	151	unitless
p25	Inhibition of mRNA transcription by HdmX	1.29	8.84	unitless

Input Data and Model

Following DNA damage, damage-activated kinases are activated and protein complexes are formed with the damaged DNA. One of the downstream effects is phosphorylation at the serine-15 epitope of p53. All input and output data were reported by (Wang, Wade et al. 2007). They are normalized to total p53 and then to the maximum value of this ratio as our input signal (DiStefano unpublished).

t	nSerineMCF7
0	0.03549555
60	1
120	0.8970129
180	0.65498729
240	0.410385466
300	0.32928888
360	0.38482679
420	0.24509552

INPUT: Serine-15 of p53



We used the MATLAB curve fitting tool to fit a polynomial to the Serine-15 p53 data. The best fit, shown, was a 6th degree polynomial:

$$u_{MCF7}(t) = \text{-}1.39\times10^{\text{-}14}t^6 + 1.846\times10^{\text{-}11}t^5 \ \text{-}1.001\times10^{\text{-}8}t^4 + 2.898\times10^{\text{-}6}t^3 \\ \text{-}0.0004702t^2 + 0.03574t + 0.03583 \tag{5}$$

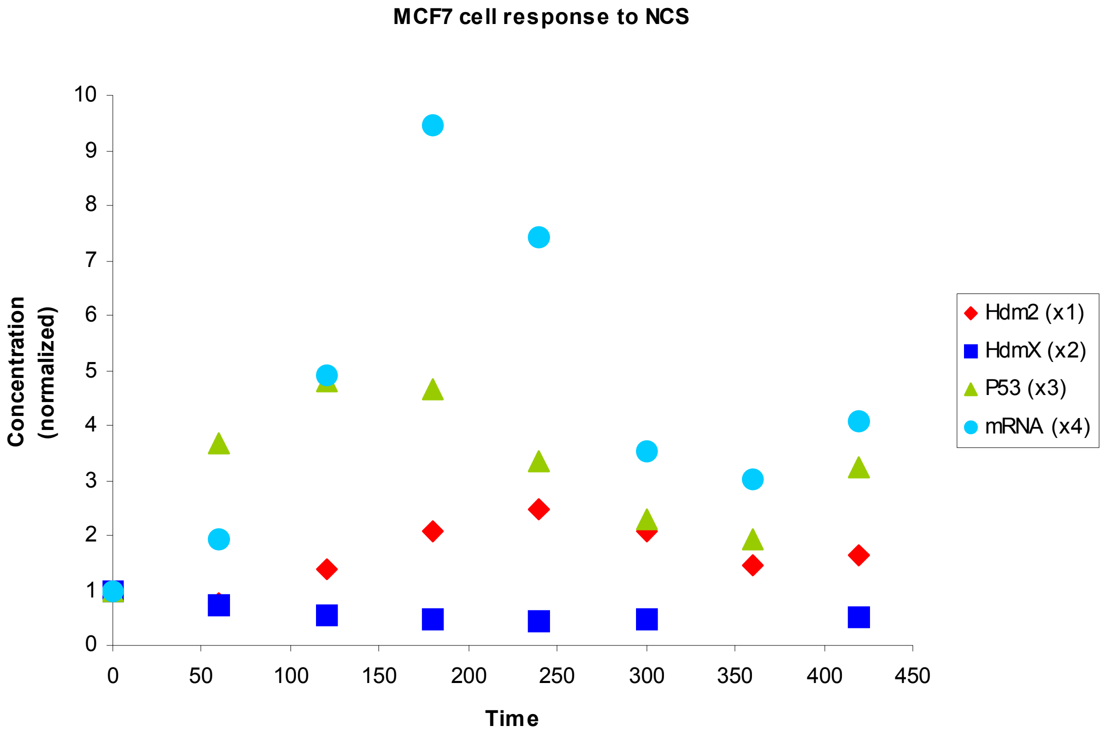
This input function corresponds to the u(t) input term in equations (1) - (4).

Output Data

MCF7 (tumorigenic mammary epithelial cells) were treated with the radiomimetic drug neocarzinostatin (NCS) to induce double stranded DNA breaks without the side effect of oxidative stress. Samples were collected every hour from t = 0 to 7. Western blot was performed for the proteins Hdm2, HdmX and p53 and measured by the LiCor system. Hdm2 mRNA was measured by quantitative PCR (Wang, Wade et al. 2007). Like the serine-15 input data, this output data is normalized to the to the total p53 and again to the maximum value of the ratio , as shown below (DiStefano unpublished).

minutes	**Hdm2**	**HdmX**	**P53**	**mRNA**
0	1	1	1	1
60	0.7698255	0.7306545	3.6565875	1.91
120	1.3891844	0.5344698	4.7947678	4.92
180	2.0677659	0.460153	4.6422326	9.47
240	2.4781249	0.4504525	3.354605	7.43
300	2.075068	0.4606971	2.2885615	3.51
360	1.4535627		1.9228018	3.02
420	1.628092	0.5065996	3.2462706	4.06

DATA: measured from mcf7 cells in response to NCS
All SDs estimated as constant = 0.25 for all output variables



III. Mathematical Methods and Definitions

Parameter Estimation

Local search using SAAMII software was inadequate to the task of generating solutions for the overparameterized p53 dynamics model (DiStefano unpublished). We turned to global and local search algorithms in several other packages to address the problem. Success was achieved with simulated annealing, a global search algorithm, which allows for occasional acceptance of uphill (against the gradient) states, allowing minimums outside of local minimums to be found (Sun, Garibaldi et al. 2012).

Parameter search by simulated annealing for our model took up to 10 hours per fitting attempt. We have also tried several other optimization methods (Steepest Descent, Levenberg-Marquardt, Nelder-Mead, Dynamic Hill Climbing, Evolution Strategy, Random Search) with varying degrees of success. Simulated annealing most consistently gave good fits in the shortest time while other methods often failed or found the same fit in a longer time. Local methods often failed to give statistics with any program, except for when started at parameter values given by our successful simulated annealing runs.

Parameter Estimation Covariance Matrices

COPASI software lacks functionality for specifying measurement data errors or data-weights in their parameter search routines. It does provide a covariance matrix estimate for the parameters estimated, based on computation of the Fisher information matrix (FIM), $\mathrm{COV}(p) \approx \mathrm{FIM}^{-1}$, but these are all based on the assumption of equal weights for the data. To circumvent this limitation, we scalar multiply these covariance matrix results by the known variance of the data, $\sigma^2 = 0.0625$. and calculate the new %CV accordingly (Distefano Still In Publication)

Building the Discretized Sensitivity Matrix (DSM)

Most parameter reduction approaches based on sensitivity analysis involve building a discretized sensitivity matrix (DSM). The term “parameter sensitivity” describes the effects of changes in parameter values on outputs of the model (Banks, Dediu et al. 2007). An output is “more sensitive” to a parameter if perturbations to the parameter have larger effects on the output. It is “less sensitive” if perturbations of the parameter results in an indiscernible change in the output (Thomaseth and Cobelli 1999; Banks, Dediu et al. 2007; Zi, Zheng et al. 2008). For a system with state variable x_i, and parameter θ_j, we can write:

$$s_{ij}\left(t\right)=\frac{\partial x_i\left(t\right)}{\partial\theta_j} \tag{6}$$

Sensitivity functions are most readily computed by approximating finite differences:

$$s_{ij}\left(t\right)=\frac{\partial x_i\left(t\right)}{\partial\theta_j}=\frac{x_i\left(\theta_j+\Delta\theta_j,t\right)-x_i\left(\theta_j,t\right)}{\Delta\theta_j} \tag{7}$$

This is the method used most often to calculate sensitivities (Daun, Rubin et al. 2008).

Our model consists of 4 equations corresponding to the 4 state variables (x_1, x_2, x_3 and x_4) and 23 parameters (p_1, …). At each time point, we calculate the sensitivity of each output variable to each parameter (4 x 23 values). We build these into a 4 x 23 matrix and this is our sensitivity matrix (Distefano Still In Publication).

$$\begin{matrix} s_{x_1p_1} & \cdots & s_{x_1p_{23}} \\ \vdots & \ddots & \vdots \\ s_{x_4p_1} & \cdots & s_{x_4p_{23}} \end{matrix} \tag{8}$$

We build a discretized sensitivity matrix (DSM) by stacking the sensitivity matrix at each sample time point below the previous one (Distefano Still In Publication).

$$\begin{array}{ccc} s_{x_1p_1}@t_1 & \cdots & s_{x_1p_{23}}@t_1 \\ \vdots & \ddots & \vdots \\ s_{x_4p_1}@t_1 & \cdots & s_{x_4p_{23}}@t_1 \\ s_{x_1p_1}@t_2 & \cdots & s_{x_1p_{23}}@t_2 \\ \vdots & \ddots & \vdots \\ s_{x_4p_1}@t_2 & \cdots & s_{x_4p_{23}}@t_2 \end{array} \tag{9}$$

and so on…

The columns of this matrix, denoted by the vector **s$_i$** are the sensitivity vectors for each parameter (i.e. 1st column is the sensitivity vector for p$_1$). The length of this vector, calculated as

$$\left\|\vec{s}\right\| = \sqrt{{s_1}^2 + {s_2}^2 + \ldots + {s_N}^2} \tag{10}$$

describes how sensitive the model is to that particular parameter (longer length = more sensitive). The direction of the vectors describe the effect that this parameter has on the model. Two parameters with similar direction have similar effects on the model.

As we have 23 parameters (= 23 columns), 8 sampling time points (t = 0, 60, 120, 180, 240, 300, 360, 420) and 4 state variables (8 x 4= 32 rows), the resulting discretized matrix for our model is 32 x 23. We used COPASI to generate the discretized sensitivity matrix DSM, given in APPENDIX B.

IV. Software Tools

We used simulation, parameter estimation, statistics and sensitivity functionality in COPASI, AMIGO and MATLAB, as described below.

COPASI (Complex Pathway Simulator)

http://www.copasi.org/

COPASI is a software application for simulation and analysis of biochemical networks (Hoops, Sahle et al. 2006; COPASI_Development_Team 2009). In addition to simulation, COPASI can also perform a number of analyses pertinent to biochemical modeling: Steady-State analysis, Stoichiometry analysis, Metabolic Control analysis, Lyapunov Exponent calculation, Time Scale Separation analysis, parameter scan, optimization, parameter estimation and Sensitivity analysis (Hoops, Sahle et al. 2006; COPASI_Development_Team 2009). We used Time Course calculation, Parameter Estimation and Sensitivity analysis tools in COPASI.

MATLAB R2010a

http://www.mathworks.com/products/matlab/

MATLAB (MATrix LABoratory) provides a comprehensive environment for computing, programming and visualizing (MathWorks 2011). In addition to the many built in functions, the functionality of MATLAB can be increased by installation of toolboxes, packages of functions designed to solve specific classes of problems. We developed a few short routines in MATLAB and used the curve fitting and AMIGO toolboxes.

AMIGO

http://www.iim.csic.es/~amigo/

AMIGO (Advanced Model Identification using Global Optimization) is a MATLAB toolbox, freely available to academic users. Installation is simple: unzip the file into your MATLAB directory, and add the folders (and all subfolders) to the MATLAB path. To run, type "AMIGO_Startup" into the command window and all the paths will be linked automatically. We found AMIGO to be a good alternative to SBML_SAT (Zi, Zheng et al. 2008) which is difficult to install and limited in application and options.

AMIGO can be used for model simulation, sensitivity analysis and ranking of parameters (both globally and locally), parameter estimation, identifiability analysis and optimal experiment design. AMIGO incorporates many numerical methods for simulation and optimization, plus multi-start and global algorithms. While the model definition process is more difficult than other software, the flexibility gained is well worth the effort. The p53 model implemented in AMIGO can be found in APPENDIX D.

COPASI vs AMIGO

We found AMIGO to be very fully featured with a lot of powerful analysis tools. However, the user unfriendly interface detracts greatly from the user experience. So for processes that COPASI and AMIGO share, we used COPASI whenever possible.

V. Parameter Reduction Algorithms

Overparametized ODE model (OPM) are characterized by unacceptably large parameter estimation variances, often with too many pairwise parameter correlations close to 1 in magnitude. Our goal is to get the best fit possible with a smaller number of parameters, each with acceptable variances. Several groups have recently developed methods to effectively reduce the number of parameters in OPMs (Brun, Kuhni et al. 2002; Smets, Bernaerts et al. 2002; Degenring, Froemel et al. 2004; Yue, Brown et al. 2006; Banks, Dediu et al. 2007; Daun, Rubin et al. 2008; De Pauw, Steppe et al. 2008; Chu and Hahn 2009; Cintron-Arias, Banks et al. 2009; Doherty and Hunt 2009; Huang, Chu et al. 2010; Quaiser, Dittrich et al. 2011). We describe these below.

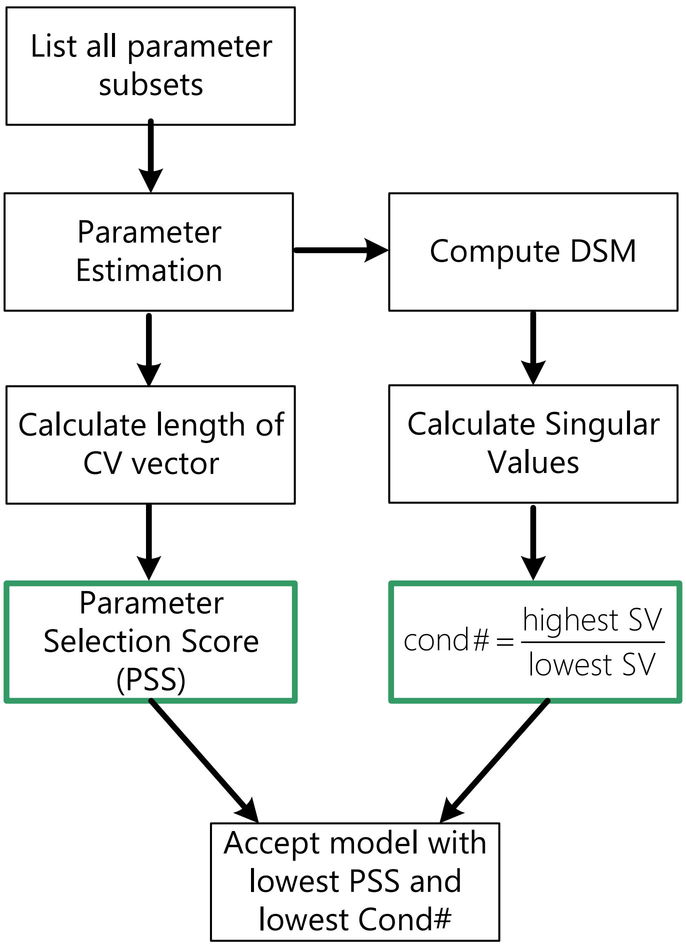
A Sensitivity Based Parameter Reduction Approach (Cintron-Arias, Banks et al. 2009)

This approach analyzes the CVs of the parameters and the sensitivity matrix generated by each possible subset of parameters. First, all possible subsets of parameters are listed. Parameter estimation is performed on each subset, fixing the parameters not in the subset to nominal values. The parameter coefficient of variations (CV) are calculated and a discretized sensitivity matrix (DSM) is generated. Using the CV vector and the DSM, the parameter selection score

(PSS) and the condition number, both defined below, are calculated. Parameter sets with the lowest PSS and condition number are recommended.

PSS is the length of the vector of parameter's CVs. For parameter sets with high uncertainty, the PSS is high, suggesting that the parameters in that set are less identifiable. Parameter sets that are considered "best" for estimation should have low PSS (low CVs).

Cintron-Arias Algorithm


List all parameter subsets
Parameter Estimation
Compute DSM
Calculate length of CV vector
Calculate Singular Values
Parameter Selection Score (PSS)
$\text{cond\#} = \frac{\text{highest SV}}{\text{lowest SV}}$
Accept model with lowest PSS and lowest Cond#

CV: Coefficient of Variation
DSM: Discretized Sensitivity Matrix

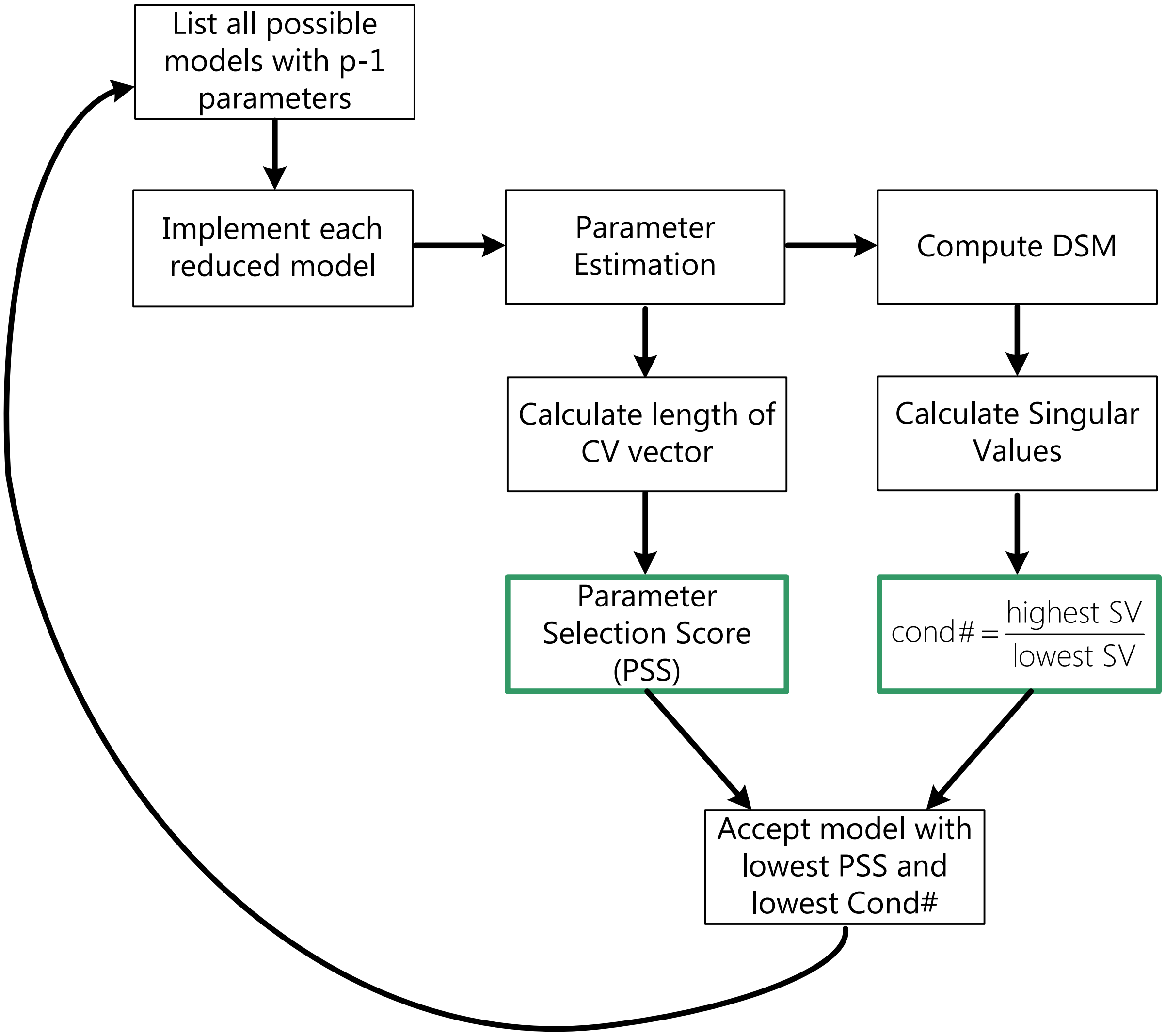
The condition number is the ratio between the highest and lowest singular values of the DSM. This is a measure of dependency within the parameter set. If the parameters are nearly dependent, the condition number of the DSM will be large. The matrix is said to be "ill-conditioned", and it will be difficult to distinguish the parameters from each other. Conversely, for sets where the parameters are more independent (more identifiable), the condition number will be small. These parameter sets are good candidates for estimation.

Sequential Adaptation of the Cintron-Arias Algorithm

In their paper, the Cintron-Arias algorithm is demonstrated on a model consisting of 3 ODEs and 11 parameters, and all reduction steps are done simultaneously on all parameters. On physical grounds, they noted that 3 parameters that must be included in the reduced parameter set, reducing the problem space from 11 to 8 parameters. Thus, 2^8 = 256 parameter sets remained to be analyzed simultaneoulsy. For our model, we have with 23 parameters – this works out to 2^23 = 8,388,608 parameter sets to consider, which is too many. For models with large numbers of parameters, it is often easier to reduce the model one parameter at a time and reevaluate the model after each removal. Such "sequential" approaches generally take less time to evaluate and are easier to implement. However, the best combination of parameters might be missed, due to parameters selected at earlier steps (Chu and Hahn 2009).

Here we modify the (Cintron-Arias, Banks et al. 2009) algorithm, so it can be applied sequentially. Instead of evaluating all possible combinations, we reduce the search space by evaluating the combinations containing P-1 parameters, where P is the number of parameters in the unreduced model. We accept the P-1 model with the lowest PSS and condition number and continue by evaluating the models with (P-1)-1 parameters. We continue until a suitable model has been reached (definitions of suitability can be adapted to each application, examples include an upper limit for the magnitude of %CVs, lower limit for singular values, etc.) Our proposed modification changes the simultaneous (Cintron-Arias, Banks et al. 2009) algorithm into a sequential process which can be completed in linear time. Code for implementation of this method can be found in APPENDIX E and F.

Sequential Adaptation of the Cintron-Arias Algorithm

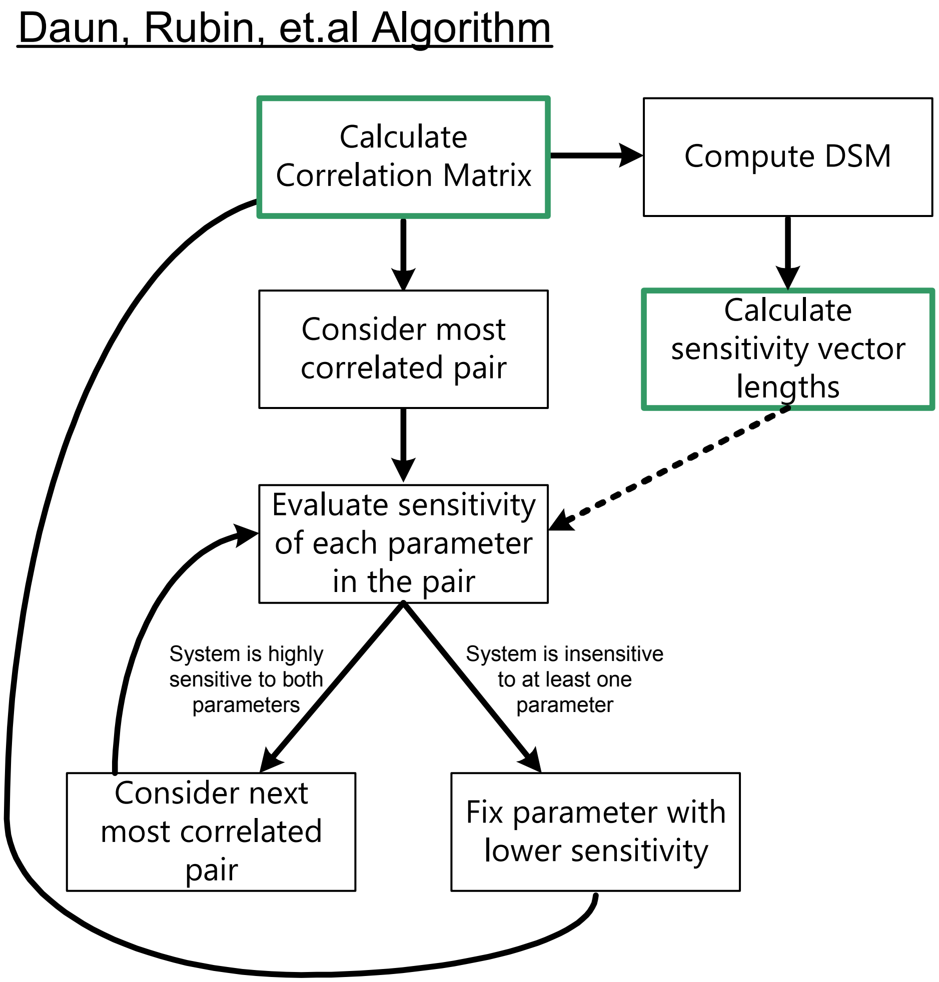


List all possible models with p-1 parameters
Implement each reduced model
Parameter Estimation
Compute DSM
Calculate length of CV vector
Calculate Singular Values
Parameter Selection Score (PSS)
$\text{cond\#} = \frac{\text{highest SV}}{\text{lowest SV}}$
Accept model with lowest PSS and lowest Cond#

CV: Coefficient of Variation
DSM: Discretized Sensitivity Matrix

Another Sequential Method

(Daun, Rubin et al. 2008; De Pauw, Steppe et al. 2008) use another sequential approach to reduce overparametized models. The two methods are quite similar and both are based on analysis of the correlation matrix and individual parameter sensitivities. These algorithms prioritize parameters with strong pairwise correlations (difficult to distinguish) that have minimal effects on the output (low sensitivity) as candidates for reduction. They differ essentially in the order of operations. In (Daun, Rubin et al. 2008), parameter correlations are computed first. Starting from the most correlated pair of parameters, the sensitivity of each parameter is evaluated by finding the length of the sensitivity vector, using equation (10). If the model is highly sensitive to both parameters, the next pair is considered. If the model is not sensitive to at least one of the parameters, the one with lower sensitivity is fixed, parameter estimation is repeated, and a new correlation matrix is calculated. This continues until the stopping criteria is reached. The stopping criterion should be adjusted based on the problem. Possible stopping criteria include: a target number of parameters to be reduced, a lower bound on the length of the sensitivity vector, a upper bound on the correlation between two parameters, etc.


Daun, Rubin, et.al Algorithm
Calculate Correlation Matrix
Compute DSM
Consider most correlated pair
Calculate sensitivity vector lengths
Evaluate sensitivity of each parameter in the pair
System is highly sensitive to both parameters
System is insensitive to at least one parameter
Consider next most correlated pair
Fix parameter with lower sensitivity

DSM: Discretized Sensitivity Matrix

A Nonsequential Method (Chu and Hahn 2009; Huang, Chu et al. 2010)

The parameter reduction scheme used in these papers is based on the discretized sensitivity matrix. This algorithm is surprisingly fast. Parameters are sorted into a predetermined number of groups, and the best candidates for estimation in each group are chosen, without listing all the possible parameter combinations. This method can quickly reduce a model with 100+ parameters into 10 or 20 parameters.

To start, (Chu and Hahn 2009) compute the DSM for the unreduced model. They perform singular value decomposition on the DSM, and use the singular values to decide how many parameters (N_g) are to be estimated – the criteria for this is adjusted to match the dimensionality of the problem. A sample criteria could be "count the number of singular values which are above 5% of the largest singular value. They calculate the length of each sensitivity vector using equation (10) on the columns of the DSM. Parameters with short vectors (e.g. <5% of the largest one), are fixed. For the remaining parameters, the "similarity measure" (see equation (11), defined below) by finding the angle between each parameter (i.e. p1 and p2, p1 and p3, etc.) "Similar" parameters are clustered into N_g groups, and from each group, the parameter with the largest sensitivity vector to chosen to be representative of the group. As this is a simultaneous method, there are no looping steps – this is the end of the process.

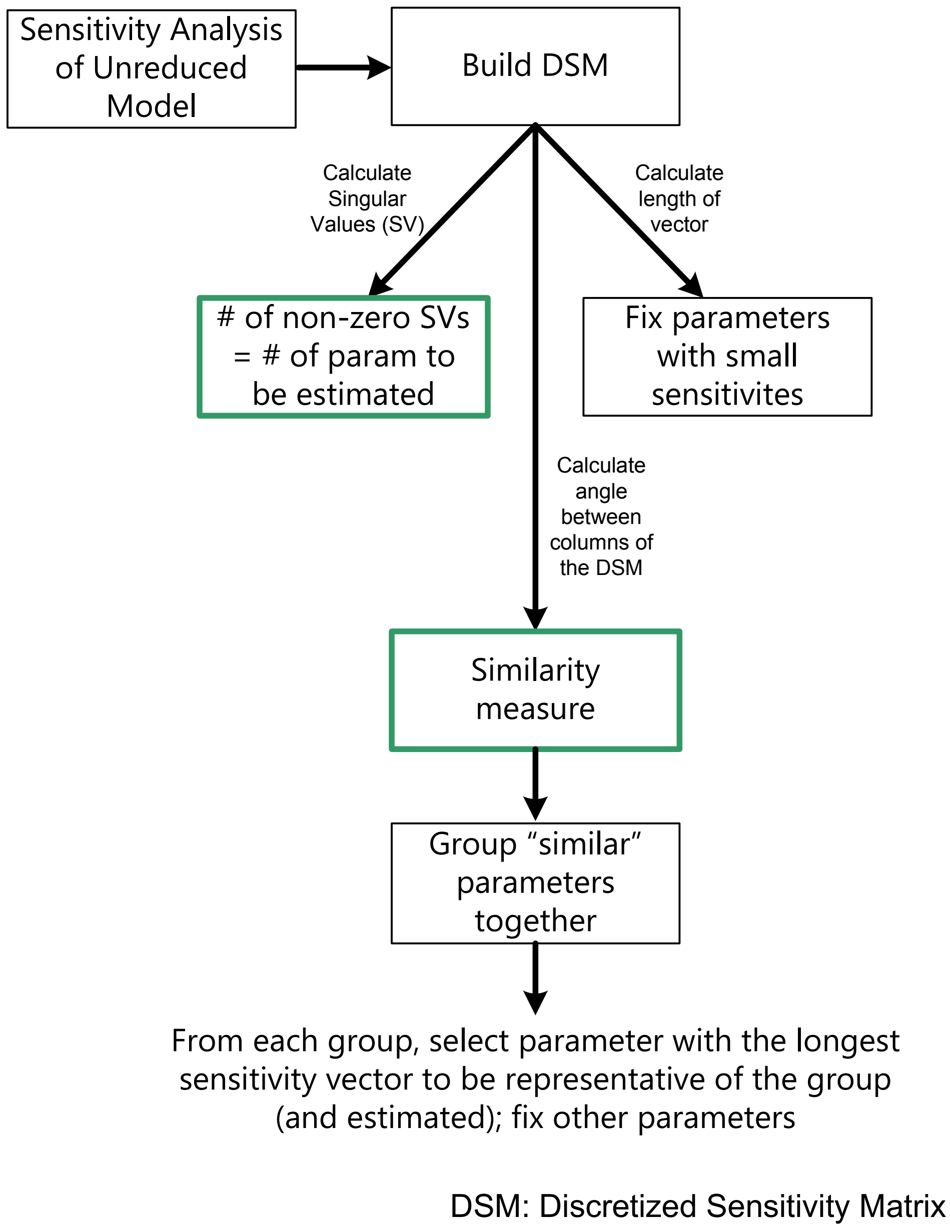
Similarity Measure (Chu and Hahn 2009)

The similarity measure defined by these authors is calculated by finding the cosine of the angle formed between two sensitivity vectors s_i and s_k:

$$\cos\phi_{ik} = \frac{\left|s_i^T s_k\right|}{\left\|s_i\right\|_2 \left\|s_k\right\|_2} \tag{11}$$

Values close to 1 means that the angle between them is small, and the two parameters are very dependent. Values close to 0 means that the angle between them is close to 90° (perpendicular), and are easily discerned from each other.

Chu, Hahn Algorithm

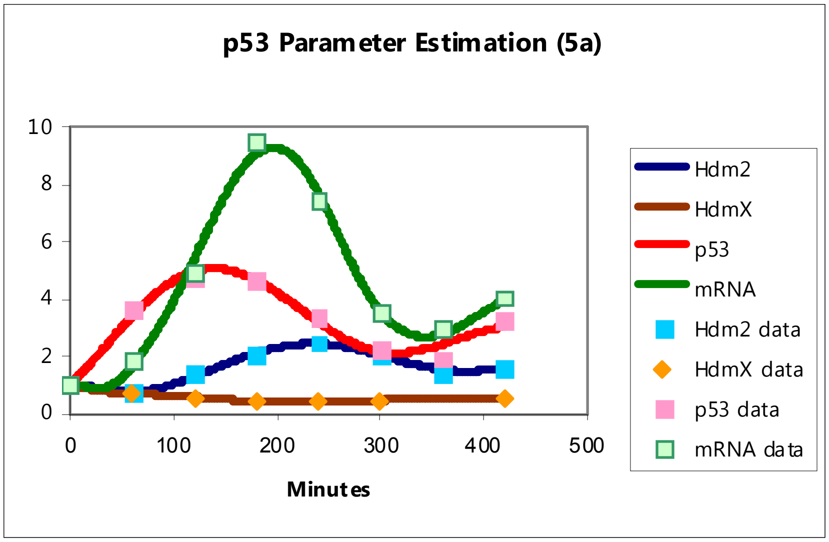

Sensitivity Analysis of Unreduced Model
Build DSM
Calculate Singular Values (SV)
Calculate length of vector
# of non-zero SVs = # of param to be estimated
Fix parameters with small sensitivites
Calculate angle between columns of the DSM
Similarity measure
Group "similar" parameters together
From each group, select parameter with the longest sensitivity vector to be representative of the group (and estimated); fix other parameters
DSM: Discretized Sensitivity Matrix

We applied these algorithms – the modified (Cintron-Arias, Banks et al. 2009) algorithm, the (Chu and Hahn 2009; Huang, Chu et al. 2010) algorithm and the (Daun, Rubin et al. 2008) algorithm – to the overparametized candidate model of tumor suppressor protein p53 dynamics. We found that fixing three parameters, p3 or p4, p17 and p23, gives the same goodness of fit for the model. The resulting reduced model has an approximately 3-fold decrease in parameter estimation errors.

IV. Results

Best Fit

The best fit of the complete model (23 parameters) to the data, shown below (without SDs =0.25 illustrated on the graph), was achieved with COPASI using the global search scheme, simulated annealing. Search parameters were: Start Temp = 1, Cooling Factor = 0.85, Tolerance = 1e-6. Total CPU time for fitting was ~ 10 hours. Resulting objective Value = 1.88.   Local multistart methods using AMIGO were also run, using the final global search parameters as ICs, without achieving a better fit.



Data for fitting was from (Wang, Wade et al. 2007), described earlier. Initial starting parameter values were adapted from the best values from a SAAM II fitting run with slightly different (simplified) equations from the current model (File name = P53 MCF7 6-30-10 - reconnected and refitted-joe-SD=0.25.stu). Adaptations were calculations of parameter values previously combined and now separated for analysis of the full model.

Results of 3 Parameter Reduction Techniques

We used 3 different model reduction methods for simplifying the overparameterized p53 dynamics model.

Adapted (Cintron-Arias, Banks et al. 2009) method resulted in fixing p23, p17, p4

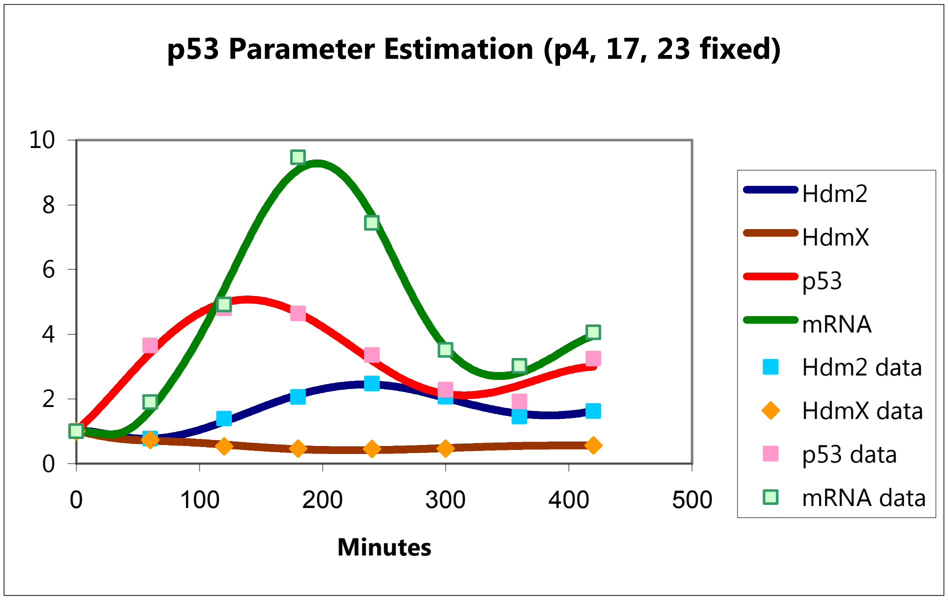
(Daun, Rubin et al. 2008) method resulted in fixing p17, p3, p23

(Chu and Hahn 2009) method resulted in fixing p3, p17, p23

The (Chu and Hahn 2009) and (Daun, Rubin et al. 2008) algorithms recommend the same parameters for model reduction.

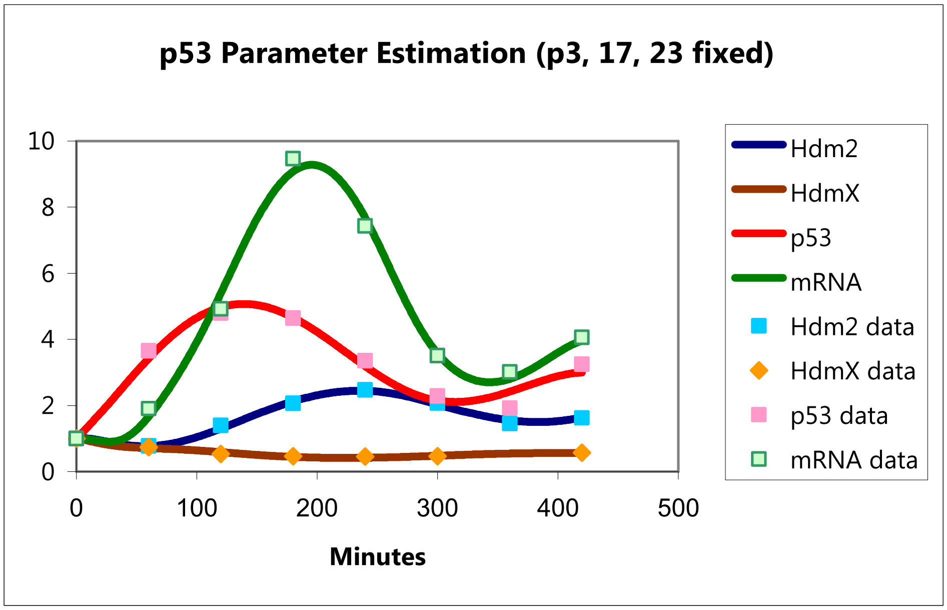
The 2 reduced models result in nearly identical fits to the data:

(Cintron-Arias, Banks et al. 2009)



(Daun, Rubin et al. 2008; De Pauw, Steppe et al. 2008)

(Chu and Hahn 2009; Huang, Chu et al. 2010)



All 3 methods resulted in fixing p17 and p23. p17 is the hill K for p53 kinase phosphorylation. p23 is the stimulation of mRNA transcription by phosphorylation of p53. These values were computed as 0 or very close to 0 in the parameter estimation, thus the equations are simplified as follows:

$$\frac{dx_1}{dt} = p_1 x_4(t-\tau) - p_3 x_1 - p_4\left(\frac{x_1^2}{p_5 + x_1}\left(1 + \frac{p_6 u}{p_7 + u}\right)\right) \tag{12}$$

$$\frac{dx_2}{dt} = p_8 - p_9 x_2 - p_{10}\left(\frac{x_1 x_2}{p_{11} + x_2}\left(1 + \frac{p_{12} u}{p_{13} + u}\right)\right) \tag{13}$$

$$\frac{dx_3}{dt} = p_{14} - p_{15} x_3 - p_{16} x_1 (1 - p_{18} u) \tag{14}$$

$$\frac{dx_4}{dt} = p_{19} - p_{20}x_4 + p_{21}\left(\frac{x_3^n}{p_{22}^n + x_3^n}\right)(1 - p_{24}x_1)(1 - p_{25}x_2) \tag{15}$$

For comparison, the original equations are:

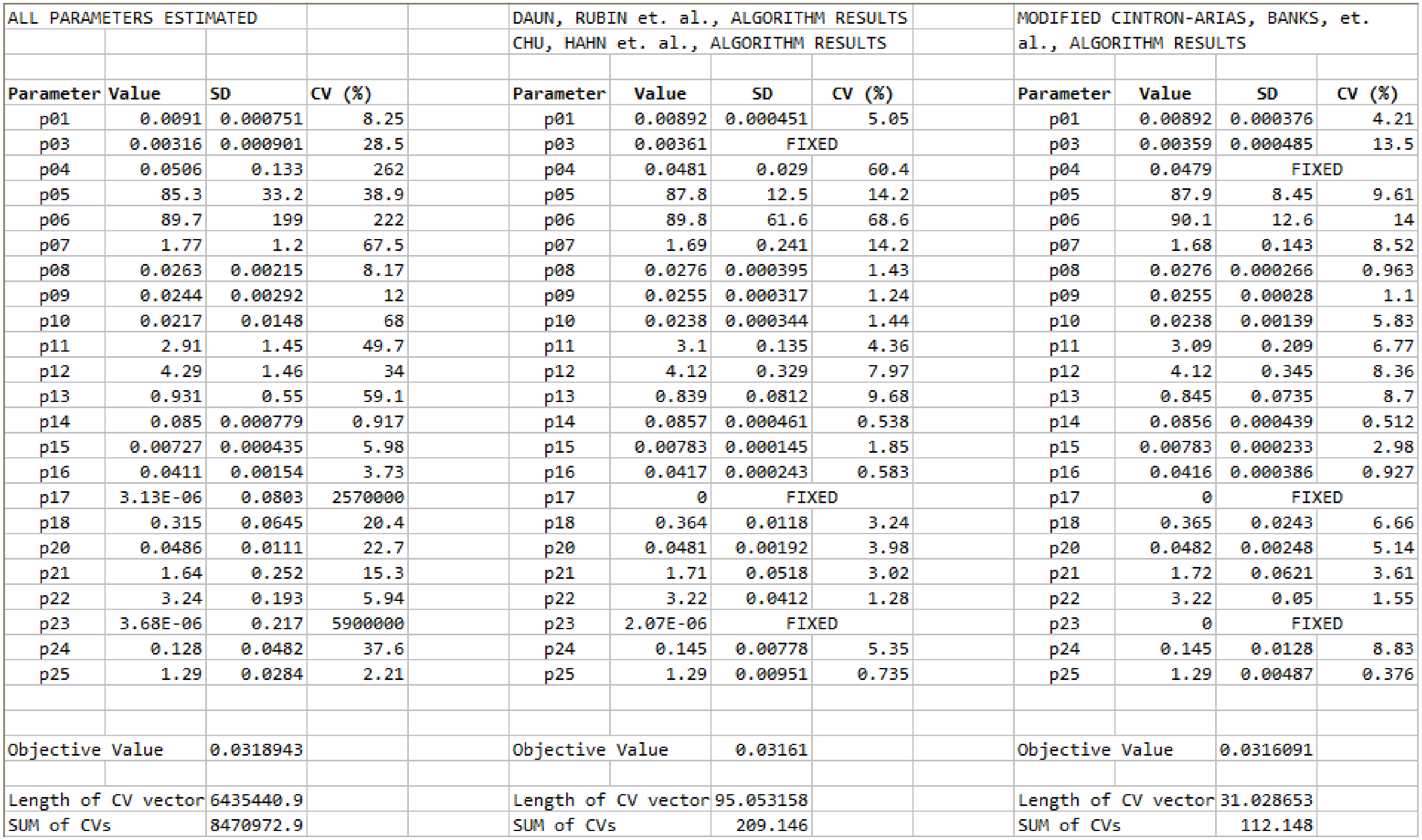
$$\frac{dx_1}{dt} = p_1 x_4(t-\tau) - p_3 x_1 - p_4\left(\frac{x_1^2}{p_5 + x_1}\left(1 + \frac{p_6 u}{p_7 + u}\right)\right) \tag{1}$$

$$\frac{dx_2}{dt} = p_8 - p_9 x_2 - p_{10}\left(\frac{x_1 x_2}{p_{11} + x_2}\left(1 + \frac{p_{12} u}{p_{13} + u}\right)\right) \tag{2}$$

$$\frac{dx_3}{dt} = p_{14} - p_{15}x_3 - p_{16}\frac{x_1 x_3}{p_{17} + x_3}(1 - p_{18}u) \tag{3}$$

$$\frac{dx_4}{dt} = p_{19} - p_{20}x_4 + p_{21}\left(\frac{x_3^n}{p_{22}^n + x_3^n}\right)(1 + p_{23}u)(1 - p_{24}x_1)(1 - p_{25}x_2) \tag{4}$$

Final parameter estimates and their variabilities are given in the table below.



ALL PARAMETERS ESTIMATED					DAUN, RUBIN et. al., ALGORITHM RESULTS					MODIFIED CINTRON-ARIAS, BANKS, et.			
					CHU, HAHN et. al., ALGORITHM RESULTS					al., ALGORITHM RESULTS			
Parameter	**Value**	**SD**	**CV (%)**		**Parameter**	**Value**	**SD**	**CV (%)**		**Parameter**	**Value**	**SD**	**CV (%)**
p01	0.0091	0.000751	8.25		p01	0.00892	0.000451	5.05		p01	0.00892	0.000376	4.21
p03	0.00316	0.000901	28.5		p03	0.00361	FIXED			p03	0.00359	0.000485	13.5
p04	0.0506	0.133	262		p04	0.0481	0.029	60.4		p04	0.0479	FIXED	
p05	85.3	33.2	38.9		p05	87.8	12.5	14.2		p05	87.9	8.45	9.61
p06	89.7	199	222		p06	89.8	61.6	68.6		p06	90.1	12.6	14
p07	1.77	1.2	67.5		p07	1.69	0.241	14.2		p07	1.68	0.143	8.52
p08	0.0263	0.00215	8.17		p08	0.0276	0.000395	1.43		p08	0.0276	0.000266	0.963
p09	0.0244	0.00292	12		p09	0.0255	0.000317	1.24		p09	0.0255	0.00028	1.1
p10	0.0217	0.0148	68		p10	0.0238	0.000344	1.44		p10	0.0238	0.00139	5.83
p11	2.91	1.45	49.7		p11	3.1	0.135	4.36		p11	3.09	0.209	6.77
p12	4.29	1.46	34		p12	4.12	0.329	7.97		p12	4.12	0.345	8.36
p13	0.931	0.55	59.1		p13	0.839	0.0812	9.68		p13	0.845	0.0735	8.7
p14	0.085	0.000779	0.917		p14	0.0857	0.000461	0.538		p14	0.0856	0.000439	0.512
p15	0.00727	0.000435	5.98		p15	0.00783	0.000145	1.85		p15	0.00783	0.000233	2.98
p16	0.0411	0.00154	3.73		p16	0.0417	0.000243	0.583		p16	0.0416	0.000386	0.927
p17	3.13E-06	0.0803	2570000		p17	0	FIXED			p17	0	FIXED	
p18	0.315	0.0645	20.4		p18	0.364	0.0118	3.24		p18	0.365	0.0243	6.66
p20	0.0486	0.0111	22.7		p20	0.0481	0.00192	3.98		p20	0.0482	0.00248	5.14
p21	1.64	0.252	15.3		p21	1.71	0.0518	3.02		p21	1.72	0.0621	3.61
p22	3.24	0.193	5.94		p22	3.22	0.0412	1.28		p22	3.22	0.05	1.55
p23	3.68E-06	0.217	5900000		p23	2.07E-06	FIXED			p23	0	FIXED	
p24	0.128	0.0482	37.6		p24	0.145	0.00778	5.35		p24	0.145	0.0128	8.83
p25	1.29	0.0284	2.21		p25	1.29	0.00951	0.735		p25	1.29	0.00487	0.376
Objective Value		0.0318943			Objective Value		0.03161			Objective Value		0.0316091	
Length of CV vector		6435440.9			Length of CV vector		95.053158			Length of CV vector		31.028653	
SUM of CVs		8470972.9			SUM of CVs		209.146			SUM of CVs		112.148	

VI. Discussion

We explored 3 methods for parameter subset selection in an overparametized model of p53/Hdm2/HdmX dynamics. Analyzing candidate parameters sets using information from the discretized sensitivity matrix, correlation matrix, similarity matrix and coefficients of variation, these methods improve the quality of the model by reducing the uncertainty in the parameter estimation process.

A shortfall of these methods is that the results are specific to the current nominal values of the parameters – if we find that the true parameter values are different from what we have reported here, it is likely that the "best" parameter subset would change. Before finding the current solution, we had invested at least 200 hours of computational effort for parameter estimation, in global searches and local multistart methods.

From the original model containing 23 parameters, we propose two candidate models of 20 parameters each. Each gives a large reduction in parameter variability measures, while maintaining the same objective value (i.e. an equally good fit to the data).

We emphasize that the following physiological interpretations of our results are for MCF7 cancer cells only.

With p17 = 0, the hill function in Eqn. (3) is reduced to a linear term dependent only on x1, or Hdm2, and not on x3 or p53 as in the original equations. This implies that the rate of degradation of p53 by Hdm2 is linearly proportional to Hdm2 in these MCF7 cancer cells – not necessarily in normal cells, not fully analyzed yet.

With p23 = 0, the term $(1+p_{23}u)$ drops out in the mRNA equation, removing any direct effect of the input u on x_4 (mRNA). This suggests that effects of DNA damage on mRNA via kinase are indirect via couplings with Hdm2, HdmX and p53.

These interpretations may be different when the normal WS1 cells are fully analyzed.

Nonzero Ambiguities: The two resulting reduced models differ by whether p3 or p4 is estimated and the other fixed. p3 is the Hdm2 basal degradation rate, and p4 is the Hdm2 degradation rate by ubiquitination. Both parameters are estimated as nonzero and thus both significantly affect Hdm2 degradation. We conclude that the relative effects of these two mechanistic pathways cannot be distinguished from the (Wang, Wade et al. 2007) data.

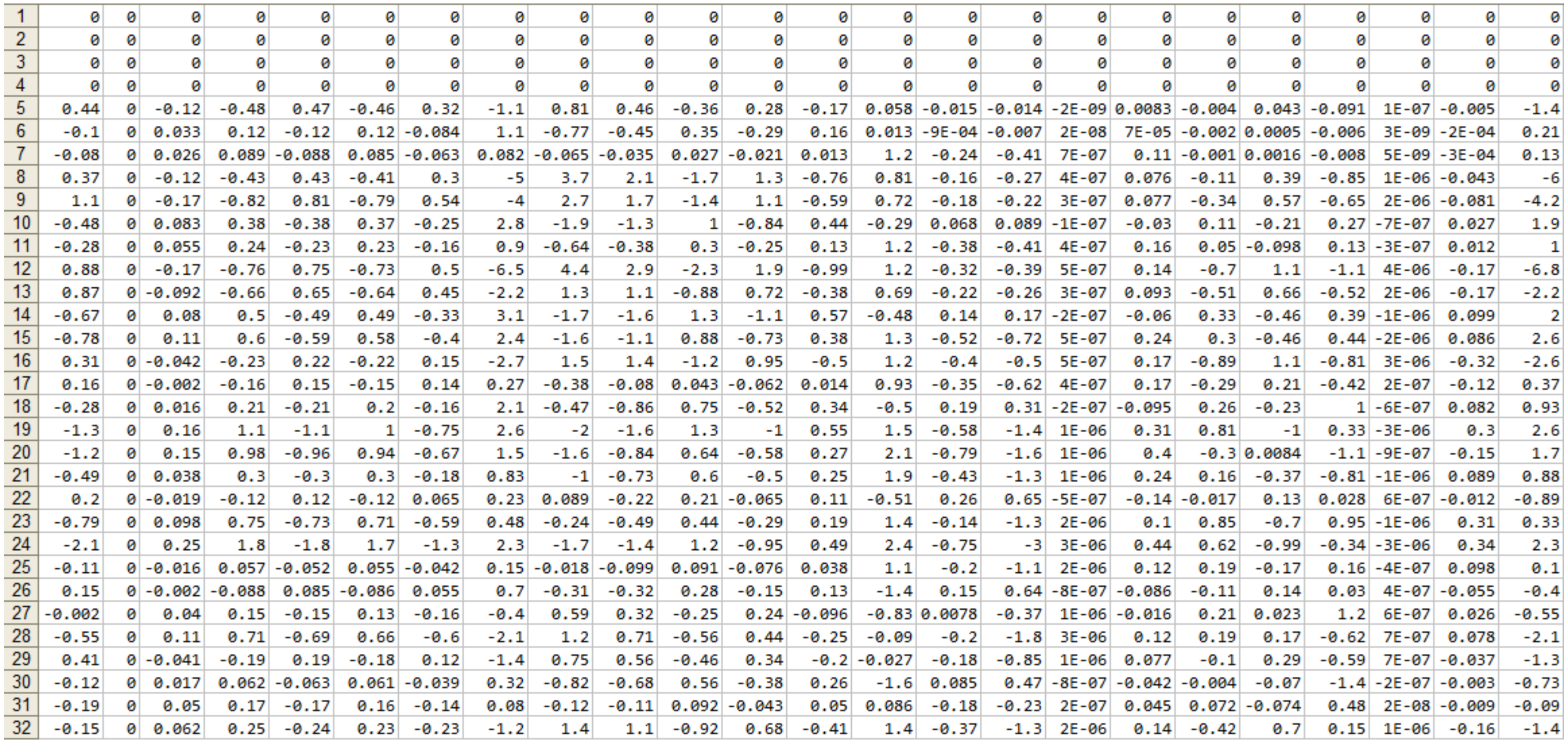
Files used to generate these results can be supplied, please write to csin@ucla.edu.

APPENDIX A: Parameter Space Reduction in the Recent Literature

Author / Year / Model	Reduction Scheme	Stop Criterion
(Quaiser, Dittrich et al. 2011) JAK-STAT	1. Calculate Hessian Matrix 2. Identify smallest eigenvalue of H 3. Search for largest component in corresponding eigenvector 4. Fixes the corresponding parameter to best estimated value (Similar to (Degenring, Froemel et al. 2004)'s METHOD 1) 5. Repeat 1-4 until all parameters are fixed. The order that the parameters were fixed is the order that the parameters are identifiable. 6. Evaluate each parameter in order from the list developed in 5), check to see if the variance is > v_p (a number you choose) If so, go to step 7. If variance is < v_p for all parameters, terminate 7. Simplify parts of model that involve the least identifiable parameters.	Continue until parameter estimation step results in SD < 1% for all estimated parameters
(Huang, Chu et al. 2010) (Chu and Hahn 2009) IL-6 signaling	1. Calculate sensitivity vectors of outputs wrt the parameters 2. Perform SVD on the sensitivity matrix 3. Determine minimum number of parameters (N_S) estimated 4. If a parameter's sensitivity vector has small length (e.g. <5% of the largest one), fix them to the nominal values 5. Cluster the parameters into N_G (where N_G is $\geq N_S$) groups by clustering parameters with similar "similarity measure" together Similarity is calculated by finding the angle between two sensitivity vectors. A value near 1 means that the two parameters have very similar effects on the OP. A value near 0 means the two parameters are close to orthogonal, i.e. they should be able to be distinguished. 6. From each group, select the parameter with the largest sensitivity vector to be representative of the group. 7. Select the number of parameters N_G representatives and refit.	Minimum number of parameters selected using SVD analysis of the sensitivity matrix
(Daun, Rubin et al. 2008) Bacterial lipopoly-saccharide in rats	1. Calculate the correlation matrix 2. Rank parameter pairs in order of correlation 3. Identify one highly correlated pair - If model OP is highly sensitive to both parameters, skip and repeat step3 - If model OP is not too sensitive to at least one of the parameters, fix the less sensitive parameter to nominal value and repeat step 1.	Continue until no more highly correlated parameters to which model output is highly sensitive remain.

(De Pauw, Steppe et al. 2008) Tree water flow and storage	1. Rank parameters from highest sensitivity to lowest sensitivity 2. Perform collinearity analysis Choose lowest collinearity and highest sensitivity parameter sets	Not specified
(Schmidt, Madsen et al. 2008) Yeast glycolysis	Method inspects each individual (complex) kinetic rate reactions and simplifies them. 1. If rate expression contains a rational term (i.e. numerator/denominator), rewrite them to the vector description of the rate expression. 2. Construct linear system of equations with results from 1) and the measurement matrix (M). 3. Inspect M for linear combinations in the columns; if these exist, the original rate expression can be reduced without loss of accuracy. 4. Remove terms in rate expression which have negligible effect	Not specified
(Degenring, Froemel et al. 2004) Glucose metabolism	1. Calculate sensitivity vectors of outputs wrt the parameters 2. Calculate the eigenvalues and eigenvectors of the sensitivity matrix 3. Choose the # of parameters (p*) to be rejected - METHOD 1: Collect the p* eigenvectors corresponding to the smallest eigenvalues. The parameter with the largest component in each considered vector is marked for rejection - METHOD 2: Collect the p* eigenvectors corresponding to the smallest eigenvalues. For each parameter, calculate the sum of squares of their components in the p* vectors. Reject parameters with the highest sum of squares. - METHOD 3: Collect the p-p* eigenvectors corresponding to the largest eigenvalues. The parameter with the largest component in each of the vectors is selected to not be rejected (i.e reject all the others)	Decided in step 3, but no recommendations given by paper

APPENDIX B: Discretized Sensitivity Matrix for the Original Unreduced Model

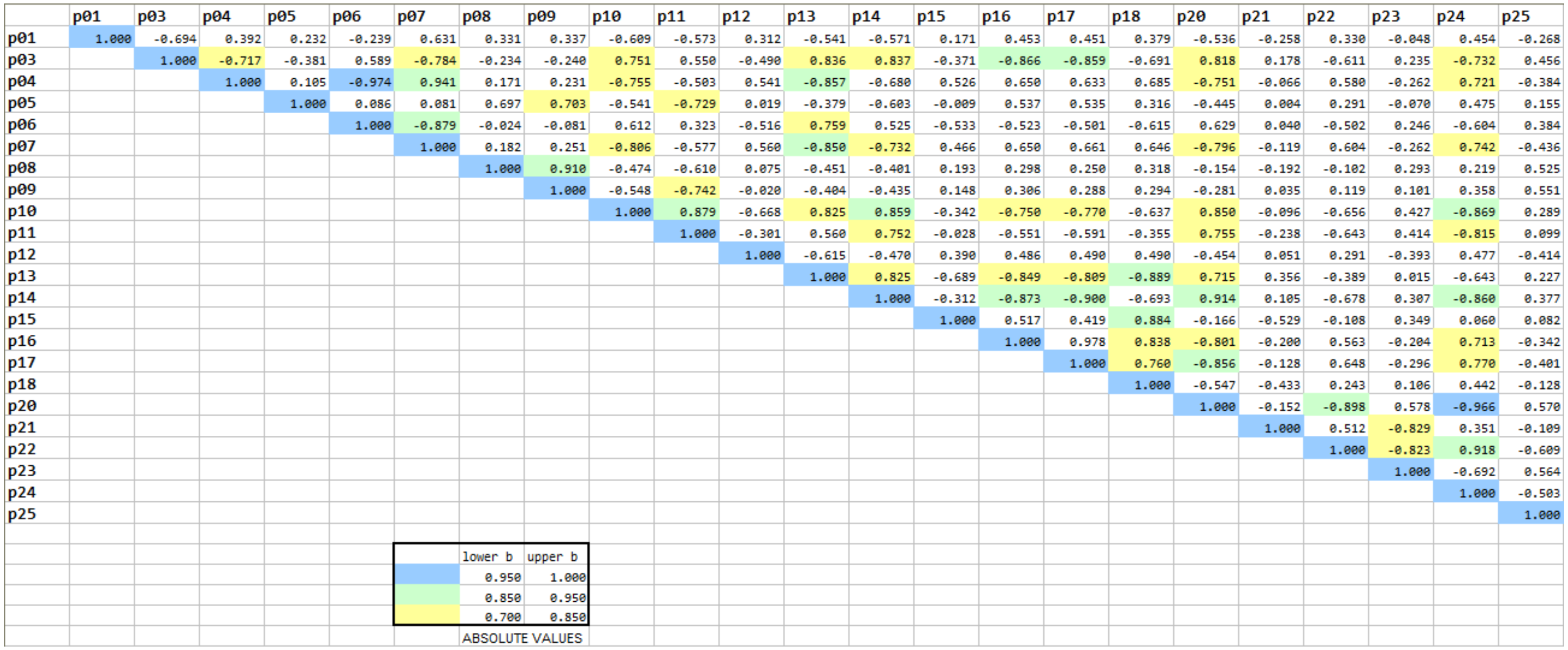


1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0.44	0	-0.12	-0.48	0.47	-0.46	0.32	-1.1	0.81	0.46	-0.36	0.28	-0.17	0.058	-0.015	-0.014	-2E-09	0.0083	-0.004	0.043	-0.091	1E-07	-0.005	-1.4
6	-0.1	0	0.033	0.12	-0.12	0.12	-0.084	1.1	-0.77	-0.45	0.35	-0.29	0.16	0.013	-9E-04	-0.007	2E-08	7E-05	-0.002	0.0005	-0.006	3E-09	-2E-04	0.21
7	-0.08	0	0.026	0.089	-0.088	0.085	-0.063	0.082	-0.065	-0.035	0.027	-0.021	0.013	1.2	-0.24	-0.41	7E-07	0.11	-0.001	0.0016	-0.008	5E-09	-3E-04	0.13
8	0.37	0	-0.12	-0.43	0.43	-0.41	0.3	-5	3.7	2.1	-1.7	1.3	-0.76	0.81	-0.16	-0.27	4E-07	0.076	-0.11	0.39	-0.85	1E-06	-0.043	-6
9	1.1	0	-0.17	-0.82	0.81	-0.79	0.54	-4	2.7	1.7	-1.4	1.1	-0.59	0.72	-0.18	-0.22	3E-07	0.077	-0.34	0.57	-0.65	2E-06	-0.081	-4.2
10	-0.48	0	0.083	0.38	-0.38	0.37	-0.25	2.8	-1.9	-1.3	1	-0.84	0.44	-0.29	0.068	0.089	-1E-07	-0.03	0.11	-0.21	0.27	-7E-07	0.027	1.9
11	-0.28	0	0.055	0.24	-0.23	0.23	-0.16	0.9	-0.64	-0.38	0.3	-0.25	0.13	1.2	-0.38	-0.41	4E-07	0.16	0.05	-0.098	0.13	-3E-07	0.012	1
12	0.88	0	-0.17	-0.76	0.75	-0.73	0.5	-6.5	4.4	2.9	-2.3	1.9	-0.99	1.2	-0.32	-0.39	5E-07	0.14	-0.7	1.1	-1.1	4E-06	-0.17	-6.8
13	0.87	0	-0.092	-0.66	0.65	-0.64	0.45	-2.2	1.3	1.1	-0.88	0.72	-0.38	0.69	-0.22	-0.26	3E-07	0.093	-0.51	0.66	-0.52	2E-06	-0.17	-2.2
14	-0.67	0	0.08	0.5	-0.49	0.49	-0.33	3.1	-1.7	-1.6	1.3	-1.1	0.57	-0.48	0.14	0.17	-2E-07	-0.06	0.33	-0.46	0.39	-1E-06	0.099	2
15	-0.78	0	0.11	0.6	-0.59	0.58	-0.4	2.4	-1.6	-1.1	0.88	-0.73	0.38	1.3	-0.52	-0.72	5E-07	0.24	0.3	-0.46	0.44	-2E-06	0.086	2.6
16	0.31	0	-0.042	-0.23	0.22	-0.22	0.15	-2.7	1.5	1.4	-1.2	0.95	-0.5	1.2	-0.4	-0.5	5E-07	0.17	-0.89	1.1	-0.81	3E-06	-0.32	-2.6
17	0.16	0	-0.002	-0.16	0.15	-0.15	0.14	0.27	-0.38	-0.08	0.043	-0.062	0.014	0.93	-0.35	-0.62	4E-07	0.17	-0.29	0.21	-0.42	2E-07	-0.12	0.37
18	-0.28	0	0.016	0.21	-0.21	0.2	-0.16	2.1	-0.47	-0.86	0.75	-0.52	0.34	-0.5	0.19	0.31	-2E-07	-0.095	0.26	-0.23	1	-6E-07	0.082	0.93
19	-1.3	0	0.16	1.1	-1.1	1	-0.75	2.6	-2	-1.6	1.3	-1	0.55	1.5	-0.58	-1.4	1E-06	0.31	0.81	-1	0.33	-3E-06	0.3	2.6
20	-1.2	0	0.15	0.98	-0.96	0.94	-0.67	1.5	-1.6	-0.84	0.64	-0.58	0.27	2.1	-0.79	-1.6	1E-06	0.4	-0.3	0.0084	-1.1	-9E-07	-0.15	1.7
21	-0.49	0	0.038	0.3	-0.3	0.3	-0.18	0.83	-1	-0.73	0.6	-0.5	0.25	1.9	-0.43	-1.3	1E-06	0.24	0.16	-0.37	-0.81	-1E-06	0.089	0.88
22	0.2	0	-0.019	-0.12	0.12	-0.12	0.065	0.23	0.089	-0.22	0.21	-0.065	0.11	-0.51	0.26	0.65	-5E-07	-0.14	-0.017	0.13	0.028	6E-07	-0.012	-0.89
23	-0.79	0	0.098	0.75	-0.73	0.71	-0.59	0.48	-0.24	-0.49	0.44	-0.29	0.19	1.4	-0.14	-1.3	2E-06	0.1	0.85	-0.7	0.95	-1E-06	0.31	0.33
24	-2.1	0	0.25	1.8	-1.8	1.7	-1.3	2.3	-1.7	-1.4	1.2	-0.95	0.49	2.4	-0.75	-3	3E-06	0.44	0.62	-0.99	-0.34	-3E-06	0.34	2.3
25	-0.11	0	-0.016	0.057	-0.052	0.055	-0.042	0.15	-0.018	-0.099	0.091	-0.076	0.038	1.1	-0.2	-1.1	2E-06	0.12	0.19	-0.17	0.16	-4E-07	0.098	0.1
26	0.15	0	-0.002	-0.088	0.085	-0.086	0.055	0.7	-0.31	-0.32	0.28	-0.15	0.13	-1.4	0.15	0.64	-8E-07	-0.086	-0.11	0.14	0.03	4E-07	-0.055	-0.4
27	-0.002	0	0.04	0.15	-0.15	0.13	-0.16	-0.4	0.59	0.32	-0.25	0.24	-0.096	-0.83	0.0078	-0.37	1E-06	-0.016	0.21	0.023	1.2	6E-07	0.026	-0.55
28	-0.55	0	0.11	0.71	-0.69	0.66	-0.6	-2.1	1.2	0.71	-0.56	0.44	-0.25	-0.09	-0.2	-1.8	3E-06	0.12	0.19	0.17	-0.62	7E-07	0.078	-2.1
29	0.41	0	-0.041	-0.19	0.19	-0.18	0.12	-1.4	0.75	0.56	-0.46	0.34	-0.2	-0.027	-0.18	-0.85	1E-06	0.077	-0.1	0.29	-0.59	7E-07	-0.037	-1.3
30	-0.12	0	0.017	0.062	-0.063	0.061	-0.039	0.32	-0.82	-0.68	0.56	-0.38	0.26	-1.6	0.085	0.47	-8E-07	-0.042	-0.004	-0.07	-1.4	-2E-07	-0.003	-0.73
31	-0.19	0	0.05	0.17	-0.17	0.16	-0.14	0.08	-0.12	-0.11	0.092	-0.043	0.05	0.086	-0.18	-0.23	2E-07	0.045	0.072	-0.074	0.48	2E-08	-0.009	-0.09
32	-0.15	0	0.062	0.25	-0.24	0.23	-0.23	-1.2	1.4	1.1	-0.92	0.68	-0.41	1.4	-0.37	-1.3	2E-06	0.14	-0.42	0.7	0.15	1E-06	-0.16	-1.4

Above is the DSM for the full unreduced p53 model.

Parameter values are nominal values obtained from the fit described on page 23.

APPENDIX C: Correlation Matrix for the Original Unreduced Model



	p01	p03	p04	p05	p06	p07	p08	p09	p10	p11	p12	p13	p14	p15	p16	p17	p18	p20	p21	p22	p23	p24	p25
p01	1.000	-0.694	0.392	0.232	-0.239	0.631	0.331	0.337	-0.609	-0.573	0.312	-0.541	-0.571	0.171	0.453	0.451	0.379	-0.536	-0.258	0.330	-0.048	0.454	-0.268
p03		1.000	-0.717	-0.381	0.589	-0.784	-0.234	-0.240	0.751	0.550	-0.490	0.836	0.837	-0.371	-0.866	-0.859	-0.691	0.818	0.178	-0.611	0.235	-0.732	0.456
p04			1.000	0.105	-0.974	0.941	0.171	0.231	-0.755	-0.503	0.541	-0.857	-0.680	0.526	0.650	0.633	0.685	-0.751	-0.066	0.580	-0.262	0.721	-0.384
p05				1.000	0.086	0.081	0.697	0.703	-0.541	-0.729	0.019	-0.379	-0.603	-0.009	0.537	0.535	0.316	-0.445	0.004	0.291	-0.070	0.475	0.155
p06					1.000	-0.879	-0.024	-0.081	0.612	0.323	-0.516	0.759	0.525	-0.533	-0.523	-0.501	-0.615	0.629	0.040	-0.502	0.246	-0.604	0.384
p07						1.000	0.182	0.251	-0.806	-0.577	0.560	-0.850	-0.732	0.466	0.650	0.661	0.646	-0.796	-0.119	0.604	-0.262	0.742	-0.436
p08							1.000	0.910	-0.474	-0.610	0.075	-0.451	-0.401	0.193	0.298	0.250	0.318	-0.154	-0.192	-0.102	0.293	0.219	0.525
p09								1.000	-0.548	-0.742	-0.020	-0.404	-0.435	0.148	0.306	0.288	0.294	-0.281	0.035	0.119	0.101	0.358	0.551
p10									1.000	0.879	-0.668	0.825	0.859	-0.342	-0.750	-0.770	-0.637	0.850	-0.096	-0.656	0.427	-0.869	0.289
p11										1.000	-0.301	0.560	0.752	-0.028	-0.551	-0.591	-0.355	0.755	-0.238	-0.643	0.414	-0.815	0.099
p12											1.000	-0.615	-0.470	0.390	0.486	0.490	0.490	-0.454	0.051	0.291	-0.393	0.477	-0.414
p13												1.000	0.825	-0.689	-0.849	-0.809	-0.889	0.715	0.356	-0.389	0.015	-0.643	0.227
p14													1.000	-0.312	-0.873	-0.900	-0.693	0.914	0.105	-0.678	0.307	-0.860	0.377
p15														1.000	0.517	0.419	0.884	-0.166	-0.529	-0.108	0.349	0.060	0.082
p16															1.000	0.978	0.838	-0.801	-0.200	0.563	-0.204	0.713	-0.342
p17																1.000	0.760	-0.856	-0.128	0.648	-0.296	0.770	-0.401
p18																	1.000	-0.547	-0.433	0.243	0.106	0.442	-0.128
p20																		1.000	-0.152	-0.898	0.578	-0.966	0.570
p21																			1.000	0.512	-0.829	0.351	-0.109
p22																				1.000	-0.823	0.918	-0.609
p23																					1.000	-0.692	0.564
p24																						1.000	-0.503
p25																							1.000

	lower b	upper b
🟦	0.950	1.000
🟩	0.850	0.950
🟨	0.700	0.850

ABSOLUTE VALUES

Above is the Correlation Matrix for the full unreduced p53 model.

Parameter values are nominal values obtained from the fit described on page 23.

APPENDIX D: AMIGO code for original p53 model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TITLE: model using 2011.10.26 equations
%
% NOTE: code structure based on "The Hodgkin-Huxley model" given in AMIGO
examples
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%============================
% RESULTS PATHS RELATED DATA
%============================
results.pathd.results_folder='p53model20120409';
    % Results in:  ...AMIGO_R2011a\Results\p53model
results.pathd.short_name='AMIGO_model';
    % figures / reports will be titled with this string
results.pathd.runident='20120421a';
    % update to keep current results from previous results



%============================
% MODEL RELATED DATA
%============================
inputs.model.input_model_type='charmodelM';
    % choices: charmodelF, charmodelM, matlabmodel, sbmlmodel,
    %          fortrammodel, blackboxmodel, blackboxcost
    % 64bit system cannot use fortran models
inputs.model.n_st=4;                % Number of states
inputs.model.n_par=25;              % Number of model parameters
inputs.model.n_stimulus=0;
    % Number of inputs, stimuli or control variables
inputs.model.names_type='custom';

inputs.model.st_names=char('x1', 'x2', 'x3', 'x4');
inputs.model.par_names=char('p1', 'p2', 'p3', 'p4', 'p5', ...
    'p6', 'p7', 'p8', 'p9', 'p10', ...
    'p11', 'p12', 'p13', 'p14', 'p15', ...
    'p16', 'p17', 'p18', 'p19', 'p20', ...
    'p21', 'p22', 'p23', 'p24', 'p25');

% Equations describing system dynamics. Time derivatives are regarded
% 'd'st_name''
inputs.model.eqns=...
        char('ip= -1.39e-14*t^6 + 1.846e-11*t^5 - 1.001e-8*t^4 + 2.898e-6*t^3
- 0.0004702*t^2 + 0.03574*t + 0.03583',...
             'dx1= p1*x4 - p3*x1 -
p4*((x1^2)/(p5+x1))*(1+(p6*ip/(p7+ip)))',...
             'dx2= p8 - p9*x2 -
p10*((x1*x2)/(p11+x2))*(1+(p12*ip/(p13+ip)))',...
             'dx3= p14 - p15*x3 - p16*((x1*x3)/(p17+x3))*(1-(p18*ip))',...
                    'p19 = p20 - p21*(1/(p22^4+1))*(1-p24)*(1-p25)',...
%exact constraint on p19
```

```
             'dx4= p19 - p20*x4 + p21*(x3^4/(p22^4+x3^4))*(1+p23*ip)*(1-
p24*x1)*(1-p25*x2)');

% Parameter starting values
inputs.model.par=[0.00909733 0 0.00316432 0.0505781 85.3301...
        89.694 1.77318 0.0262511 0.0243828 0.0217436...
        2.90862 4.29317 0.930501 0.0849594 0.00726627...
        0.0411372 3.13E-06 0.315467 0.052283251 0.0486287...
        1.64315 3.24445 3.68E-06 0.128005 1.28515];           %p21-25


%==================================
% EXPERIMENTAL SCHEME RELATED DATA
%==================================

 inputs.exps.n_exp=1;                 % #experiments = 1 (1 data set)

 inputs.exps.n_obs{1}=4;
 inputs.exps.obs_names{1}=char('obs1','obs2','obs3','obs4');
        % Name of the observed quantities per experiment

 inputs.exps.obs{1}=char('obs1=x1','obs2=x2','obs3=x3','obs4=x4');
 % all states in the model are observed

 % Initial conditions for each experiment
 inputs.exps.exp_y0{1}=[1 1 1 1];
 % Experiments duration
 inputs.exps.t_f{1}=420;
 % Number of sampling times
 inputs.exps.n_s{1}=8;
 % [] Sampling times, by default equidistant
 inputs.exps.t_s{1}=[0:60:420];


%==================================
% EXPERIMENTAL DATA RELATED INFO
%==================================

inputs.exps.data_type='real';      % Type of data:
'pseudo'|'pseudo_pos'|'real'
inputs.exps.noise_type='homo';
inputs.exps.exp_data{1}=[1 1 1 1
    0.769825528 0.730654465 3.656587533 1.91
    1.389184409 0.534469831 4.794767802 4.92
    2.067765914 0.46015303 4.642232555 9.47
    2.478124881 0.450452468 3.354605021 7.43
    2.075068041 0.460697055 2.288561511 3.51
    1.453562721 0.48 1.922801847 3.02     % NaN in 2nd column changed t0 0.48
    1.628091967 0.50659958 3.246270642 4.06];
inputs.exps.std_dev{1}=0.25;                        % sd of experimental data


%==================================
% UNKNOWNS RELATED DATA
```

```
%==================================

% GLOBAL UNKNOWNS (SAME VALUE FOR ALL EXPERIMENTS)

inputs.PEsol.id_global_theta=char('all');            % 'all'|User selected

% Minimum allowed values for the parameters
inputs.PEsol.global_theta_min=[0 0 0 0 0 ...    %p1-5
                               0 0 0 0 0 ...    %p6-10
                               0 0 0 0 0 ...    %p10-15
                               0 0 0 0 0 ...    %p16-20
                               0 0 0 0 0];      %p21-25

% Maximum allowed values for the paramters
inputs.PEsol.global_theta_max=[1 0 1 1 100 ...       %p1-5
                               100 10 1 1 1 ...     %p6-10
                               5 200 5 1 1 ...       %p11-15
                               1000 2000 2 1 100 ... %p16-20
                               10 5 1 1 10];        %p21-25

% Initial guess for parameters
inputs.PEsol.global_theta_guess=[0.00909733 0 0.00316432 0.0505781 85.3301 ...
    89.694 1.77318 0.0262511 0.0243828 0.0217436 ...
    2.90862 4.29317 0.930501 0.0849594 0.00726627 ...
    0.0411372 3.13E-06 0.315467 0.052283251 0.0486287 ...
    1.64315 3.24445 3.68E-06 0.128005 1.28515];
            % by default, AMIGO will use the mean value btw min and max as
            % the initial guess. values above are from
            % 20120409 p53 model 20111006 eqns SAAM start sim anneal 5a.cps


% GLOBAL INITIAL CONDITIONS
inputs.PEsol.id_global_theta_y0='none';     % do not estimate initial states


% LOCAL UNKNOWNS (DIFFERENT VALUES FOR DIFFERENT EXPERIMENTS)

inputs.PEsol.id_local_theta{1}='none';      % [] 'all'|User selected| 'none'
(default)
inputs.PEsol.id_local_theta_y0{1}='none';   % [] 'all'|User selected| 'none'
(default)


%==================================
% COST FUNCTION RELATED DATA
%==================================

inputs.PEsol.PEcost_type='llk';     % 'llk' (log likelihood)
inputs.PEsol.llk_type='homo_var';


%==================================
% NUMERICAL METHODS RELATED DATA
%==================================
% SIMULATION

% [] IVP solver: 'radau5'(default, fortran)|'rkf45'|'lsodes'|
%                'lsodesst'|'lsoda'|
%                'ode15s' (default, MATLAB, sbml)|'ode113'
% [] Sensitivities solver: 'odessa' (default, fortran)|
%                          'sensmat' (matlab) |
%                          'fdsens' (finite differences)
% [] IVP solver integration tolerances

inputs.ivpsol.ivpsolver='ode15s';
inputs.ivpsol.senssolver='sensmat';
inputs.ivpsol.rtol=1.0D-4;
inputs.ivpsol.atol=1.0D-4;




% OPTIMIZATION
inputs.nlpsol.nlpsolver='ssm';
    % [] NLP solver: ssm (Scatter Search for global optimization in MATLAB


%===================================
% DISPLAY OF RESULTS
%===================================

% [] Display of figures: 'full'|'medium'(default)|'min' |'noplot'
results.plotd.plotlevel='full';
% [] Figures may be saved in .eps (1) or only in .fig format (0) (default)
results.plotd.epssave=0;
% [] Maximum number of states per figure
results.plotd.number_max_states=8;
% [] Maximum number of observables per figure
results.plotd.number_max_obs=8;
% [] Number of times to be used for observables and states plots
results.plotd.n_t_plot=100;
% [] Integration tolerances for the contour plots.
results.plotd.contour_rtol=1e-7;
%    ADVISE: These tolerances should be a little bit strict
results.plotd.contour_atol=1e-7;
% [] Number of points for plotting the contours x and y direction
results.plotd.nx_contour=60;
results.plotd.ny_contour=60;                          %    ADVISE: >=50
% [] Maximum number of unknowns histograms per figure (multistart)
results.plotd.number_max_hist=8;
```

APPENDIX E: Code for Cintron et. al. Algorithm

```
% input DSM
% input parameters list
% input CV vector

%p53data20120409data5aglobal;           % CARE: if WARNING1 implemented,
comment this out.

%    check that DSM is the same width as parameters list
%    count # of columns (i.e. the # of parameters being considered)

numparam = length(parameters);

if numparam ~= length(CVs)
    error('parameters list and CVs list are not the same size');
elseif numparam ~= size(DSM, 2)
    error('DSM width is not the same size as the # of parameters');
else
    disp('data size OK');
end


%    for loop to remove column in DSM one at a time
%                remove CV value for the appropriate parameter
%                remove parameter fr parameter list one at a time
%       calc cond#
%       calc pss
%       keep track of parameter vector
%       -------------------------------------------
%       | Parameter Vector |   cond#    |    pss   |
%       -------------------------------------------
%       |     ...          |    ...     |    ...   |
%
%    result is #parameter x 3 matrix
%
% output matrix from above

% note: row x column

% create matrix for info storage

results = [];
setlist = [];

for index = 1:numparam
    % copy matrices into matrices for manipulation
    candDSM = DSM;
    candCVs = CVs;
    % remove "index" column from candidate matrices
    candDSM(:, index) = [];
    candCVs(:, index) = [];
    setlist{index, 1} = char(strcat('no_', parameters(index)));

    % make calculations and fill in results matrix
    % calculate condition # (highest sing value / lowest sing val)
    calcCond = condNum(candDSM);
    % calculate pss (magnitude of CV vector)
    calcPss = pss(candCVs);

    % fill out table and analyze
    results(index, 1) = calcCond;
    results(index, 2) = calcPss;
    results(index, 3) = calcCond + calcPss;
    results(index, 4) = calcCond*calcPss;
end


% what is minimum condition #?
[minCond, mCondLoc] = min(results(:, 1));
% what is minimum pss?
[minPss, mPssLoc] = min(results(:, 2));
% min additive
[minAdd, mAddLoc] = min(results(:, 3));
% min multiplicative
[minMult, mMultLoc] = min(results(:, 4));

% WARNING1
% if we can decide on criteria of what is the "lowest" (either by summing
% or multiplying), we can write in code here to remove the appropriate
% columns in the data, and just simply run this script again.

% Also, we should consider generating a new DSM matrix with the previously
% worst parameters fixed.


disp('Lowest Condition#')
disp(['Parameter set: ', setlist{mCondLoc}, '          line ',
num2str(mCondLoc)])
disp('Lowest Parameter Selection Score')
disp(['Parameter set: ', setlist{mPssLoc}, '          line ',
num2str(mPssLoc)])

disp('Lowest cond# + PSS')
disp(['Parameter set: ', setlist{mAddLoc}, '          line ',
num2str(mAddLoc)])
disp('Lowest cond# x PSS')
disp(['Parameter set: ', setlist{mMultLoc}, '          line ',
num2str(mMultLoc)])
```

APPENDIX F: Auxiliary functions for Cintron et. al. Algorithm

```
% pss(CVvector)
% =============

% IP:  Vector of coefficient of variances for the parameters
% OP:  Parameter Selection Score (pss) as described in the Cintron, Banks,
% Capaaldi, Lloyd 2009 paper

% input vector
%    pss = magnitude of CV vector
% output parameter selection score


function length = pss(CVvector)
% FUNCTION NAME: pss
% INPUT: Vector containing Coefficients of Variation for the parameters
% OUTPUT: Length of vector

length = norm(CVvector);
```

```
% condNum(DSMatrix)
% ================

% input sensitivity vectors
%            SVD of matrix created with vectors
%            S matrix, find smallest and largest #s
%            cond# = highest singular / lowest singular
% output condition number


function value = condNum(DSMatrix)
% FUNCTION NAME: condNum
% INPUT: Discretized Sensitivity Matrix (matrix)
% OUTPUT: condition number (double)

% calculate singular values using matlab function svd
% OP of svd is a matrix containing the singular values
singValues = svd(DSMatrix);

% calculate condition number
value = max(singValues) / min(singValues);
```

APPENDIX G: Code for Daun Rubin Algorithm

This code calculates the length of the parameter sensitivity vectors from the DSM.

```
psens = [];
DSM = DSM5a;        % discretized sensitivity matrix, preloaded into
workspace
numparam = 24;
for index = 1:numparam
    psens(index, 1) = norm(DSM(:,index),2);
end
```

APPENDIX H: Code for Chu Hahn Algorithm

```
% Load DSM
DSM = DSM5a;

% Load Parameter list
parameters = {'p1' 'p2' 'p3' 'p4' 'p5'...
    'p6' 'p7' 'p8' 'p9' 'p10'...
    'p11' 'p12' 'p13' 'p14' 'p15'...
    'p16' 'p17' 'p18' 'p20'...
    'p21' 'p22' 'p23' 'p24' 'p25'};

% Remove low sensitivity parameters from consideration
% low sens = <5% of high sens
sensLength = [];
for i = 1:size(DSM, 2)
    sensLength(i) = norm(DSM(:, i),2);
end

minSens = max(sensLength)*0.05;

i = 1;
while i < size(sensLength, 2)
    if sensLength(i) < minSens
        sensLength(i) = [];
        parameters(i) = [];
        DSM(:, i) = [];
    else
        i = i+1;
    end
end

numParam = size(DSM, 2);

% Preform SVD
[U,S,V] = svd(DSM);

% Calculate the cut off criteria for singular values
minsing = max(max(S))/1000;

% Count number of sing values > ____
% This becomes ns, the # of parameter estimated

ns = size(find(S > minsing),1);    % counts # of entries larger than minsing

% Make matrix to keep information
pAngles = zeros(numParam, numParam);

% calculating angle
% cos phi = Si Transpose Sk / ||Si||^2 * ||Sk||^2
% For loop for first parameter
for i = 1:numParam
    % For loop for 2nd parameter
    for k = i+1:numParam
```

```
        Si = DSM(:, i);
        Sk = DSM(:, k);
        % calculate angle btw sens vectors and fill in table
        pAngles(i, k) = abs(Si.'*Sk) / (norm(Si,2)^2 * norm(Sk,2)^2);
    end
end
```

References

Banks, H. T., S. Dediu, et al. (2007). "Sensitivity functions and their uses in inverse problems." Journal of Inverse and Ill-Posed Problems **15**(7): 683-708708.

Brun, R., M. Kuhni, et al. (2002). "Practical identifiability of ASM2d parameters - systematic selection and tuning of parameter subsets." Water Research **36**(16): 4113-4127.

Chu, Y. and J. Hahn (2009). "Parameter Set Selection via Clustering of Parameters into Pairwise Indistinguishable Groups of Parameters." Industrial & Engineering Chemistry Research **48**(13): 6000-6009.

Cintron-Arias, A., H. T. Banks, et al. (2009). "A sensitivity matrix based methodology for inverse problem formulation." Journal of Inverse and Ill-Posed Problems **17**(6): 545-564.

Daun, S., J. Rubin, et al. (2008). "An ensemble of models of the acute inflammatory response to bacterial lipopolysaccharide in rats: Results from parameter space reduction." Journal of Theoretical Biology **253**(4): 843-853.

De Pauw, D. J. W., K. Steppe, et al. (2008). "Identifiability analysis and improvement of a tree water flow and storage model." Mathematical Biosciences **211**(2): 314-332.

Degenring, D., C. Froemel, et al. (2004). "Sensitivity analysis for the reduction of complex metabolism models." Journal of Process Control **14**(7): 729-745.

Distefano, J. J. (Still In Publication). Dynamic Systems Biology Modeling & Simulation.

Distefano, J. J. and C. Cobelli (1980). "ON PARAMETER AND STRUCTURAL IDENTIFIABILITY - NON-UNIQUE OBSERVABILITY-RECONSTRUCTIBILITY FOR IDENTIFIABLE SYSTEMS, OTHER AMBIGUITIES, AND NEW DEFINITIONS." Ieee Transactions on Automatic Control **25**(4): 830-833.

DiStefano, J. J., III, et. al. (unpublished).

Doherty, J. and R. J. Hunt (2009). "Two statistics for evaluating parameter identifiability and error reduction." Journal of Hydrology **366**(1-4): 119-127.

Huang, Z., Y. Chu, et al. (2010). "Model simplification procedure for signal transduction pathway models: An application to IL-6 signaling." Chemical Engineering Science **65**(6): 1964-1975.

MathWorks, I. (2011). "Matlab 2012 datasheet." Retrieved May 22, 2012, from http://www.mathworks.com/tagteam/70533_91199v01_MATLABDataSheet_v9.pdf?s_cid=ML2012_bb_datasheet.

Quaiser, T., A. Dittrich, et al. (2011). "A simple work flow for biologically inspired model reduction - application to early JAK-STAT signaling." Bmc Systems Biology **5**.

Schmidt, H., M. F. Madsen, et al. (2008). "Complexity reduction of biochemical rate expressions." Bioinformatics **24**(6): 848-854.

Smets, I., K. Bernaerts, et al. (2002). "Sensitivity function-based model reduction - A bacterial gene expression case study." Biotechnology and Bioengineering **80**(2): 195-200.

Sun, J. Y., J. M. Garibaldi, et al. (2012). "Parameter Estimation Using Metaheuristics in Systems Biology: A Comprehensive Review." Ieee-Acm Transactions on Computational Biology and Bioinformatics **9**(1): 185-202.

Thomaseth, K. and C. Cobelli (1999). "Generalized sensitivity functions in physiological system identification." Annals of Biomedical Engineering **27**(5): 607-616.

Wang, Y. V., M. Wade, et al. (2007). "Quantitative analyses reveal the importance of regulated Hdmx degradation for P53 activation." Proceedings of the National Academy of Sciences of the United States of America **104**(30): 12365-12370.

Yue, H., M. Brown, et al. (2006). "Insights into the behaviour of systems biology models from dynamic sensitivty and identifiability analysis: a case study of an NF-kappa B signalling pathway." Molecular Biosystems **2**(12): 640-649.

Zi, Z., Y. Zheng, et al. (2008). "SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool." Bmc Bioinformatics **9**.