BioPSy: An SMT-based Tool for Guaranteed Parameter Set Synthesis of Biological Models

Curtis Madsen, Fedor Shmarov, and Paolo Zuliani

School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
{curtis.madsen, f.shmarov, paolo.zuliani}@ncl.ac.uk

Abstract. The *parameter set synthesis* problem consists of identifying *sets* of parameter values for which a given system model satisfies a desired behaviour. This paper presents BioPSy, a tool that performs *guaranteed* parameter set synthesis for *ordinary differential equation* (ODE) biological models expressed in the *Systems Biology Markup Language* (SBML) given a desired behaviour expressed by time-series data. Three key features of BioPSy are: 1) BioPSy computes parameter intervals, not just single values; 2) for the identified intervals the model is formally guaranteed to satisfy the desired behaviour; and 3) BioPSy can handle virtually any Lipschitz-continuous ODEs, including nonlinear ones. BioPSy is able to achieve guaranteed synthesis by utilising *Satisfiability Modulo Theory* (SMT) solvers to determine acceptable parameter intervals. We have successfully applied our tool to several biological models including a prostate cancer therapy model, a human starvation model, and a cell cycle model.

1 Introduction

Computational modelling is central to many scientific and engineering disciplines. For instance, the field of *systems biology* [17] uses modelling to gain a greater understanding of how biology works. Similarly, in *synthetic biology* [1], models are created in an attempt to engineer new, useful biological systems. This field typically develops models and analyses them *in silico* (on a computer) before synthesising the object of the model *in vitro* (in a test tube in the lab) or *in vivo* (within an organism). Biological systems in both systems biology and synthetic biology are often constructed with deterministic dynamics and can be readily translated into *ordinary differential equation* (ODE) models using *mass action kinetics*. There are many well known methods and tools for simulating ODEs that can be used to obtain results on the behaviour of the biological systems (*e.g.*, MATLAB). However, obtaining reliable results requires that all parts of a model are accurately defined. In particular, a key component to modelling biological systems is selecting the correct model parameters. Since quantitative parameters are often difficult or impossible to measure experimentally, a problem that often arises is how to select parameter values to achieve desired model behaviours. Indeed, small parameter variations can lead to vastly different results when simulating biological models.

In order to determine acceptable values for the parameters of a system, modellers have employed methods that perform *parameter synthesis*. The parameter set synthesis problem consists of determining *ranges* (intervals) of parameters for which a model's temporal behaviour remains in satisfactory states, usually described by time-series data. Formally, parameter synthesis is categorised as a *reachability problem* [2] where the solution to a set of ODEs is known for a finite number of time points, but some of the parameter values that lead to that solution are missing. For instance, the parameter, k, can be synthesised in the ODE model given by $x'(t) = kt$. Given the time-series data in which $x = \{0, 1, 4, 9\}$ for $t = \{0, 1, 2, 3\}$, it is easy to see that k should be 2. However, if the system is noisy and the values of x can vary by, say, 0.1, solving the parameter synthesis problem for k will produce an interval such as $[1.978, 2.022]$.

This paper presents `BioPSy`, a tool that performs parameter set synthesis on biological models comprised not only of mass action kinetics, but also of general Lipschitz-continuous ODEs. Models are specified using the well-known *Systems Biology Markup Language* (SBML) [14]. `BioPSy` accomplishes parameter synthesis by extracting a collection of ODEs from an SBML model and formulating these ODEs along with time-series data into a *Satisfiability Modulo Theory* (SMT) problem. It then leverages the SMT solver `dReal` [12] to incrementally narrow down the parameter search space. Given a parameter domain, precision, and time-series data expressing desired behaviour, `BioPSy` returns

- a set of feasible (acceptable) parameter ranges - these are *formally* and *numerically* guaranteed to satisfy the synthesis problem;
- a set of infeasible (unsuitable) parameter ranges - these are *formally* and *numerically* guaranteed not to satisfy the synthesis problem; and
- a set of parameter ranges where, because of the given precision, `BioPSy` is unable to determine if they satisfy the synthesis problem.

Note that, depending on the problem at hand, any of the three sets above may be empty, although not all at the same time. We remark again that `BioPSy` can handle nonlinear ODEs, and its answers have mathematical proof strength.

Related Work A simple way to perform parameter synthesis is first to discretise the parameter space (if necessary) and then use exhaustive simulation or Monte Carlo methods to determine which simulations satisfy a desired behaviour. Indeed, many tools utilise simulation-based approaches to find acceptable parameter values. For example, `COPASI` [13], a well known biochemical network simulator, uses methods such as genetic algorithms, particle swarm simulations, differential evolution, and simulated annealing among others to perform parameter estimation on SBML models. Tools like `COPASI` as well as others such as `SBML-PET` [28] can also give confidence intervals for the parameters that they estimate. Furthermore, there is a collection of applications that leverage the `MATLAB` framework to provide similar parameter estimation methods. These tools include `AMIGO` [3], a tool that uses a collection of initial value problem and non-linear optimization methods; `PottersWheel` [21], a tool that uses deterministic and stochastic optimisation techniques in concert to explore a logarithmic parameter space; and SBT [23], a tool that allows users to define their own cost-functions and use custom optimisation methods. These tools trade-off between how fine-grained the parameter search is and how much computation time is required to find acceptable parameter values.

Other approaches utilise numerical and formal methods to prove that a model meets certain criteria [4, 5, 27]. For example, Bernstein polynomials and linear programming [10], and probabilistic model checking [26] approaches can be applied to the parameter set synthesis problem. Model checking methods work by partitioning the parameter space into classes of equivalent behaviours for the various parameter values, which are then systematically validated. Simulation and model checking can be combined in a hybrid approach to efficiently search the parameter space. For example, the statistical model checking technique proposed in [16] enables parameter synthesis for stochastic biological models formulated as *continuous-time Markov chains* using temporal logic specifications (bounded LTL formulae) to express desired behaviours. Simulations can also be used to perform sensitivity analysis limiting how much of the state space the model checker will have to analyse [8]. Additionally, some methods formulate parameter synthesis as an SMT problem, but they usually handle restricted classes of models, *e.g.*, transition systems with linear dynamics [7] or with monotone dynamics [22], while we support very general dynamics such as nonlinear ODEs. Although some approaches can handle complex systems with a large number of parameters [9], their implementations are usually problem-specific.

Finally, a notion related to parameter synthesis is that of parameter identifiability, *i.e.*, whether parameters can be uniquely identified from data. This notion is usually explored in the context of specific classes of dynamics and error behaviours — see, *e.g.*, [19] and references therein.

2 Methods

We sketch the parameter synthesis technique and give implementation details of BioPSy. Full details of the theory will appear in a forthcoming paper.

Algorithm BioPSy takes as input an SBML model file, a time-series data file, a list of model parameters to synthesise with their initial ranges, a noise value (η), a precision value (δ) for the SMT solver, and a precision value (ϵ) for the parameter synthesis algorithm. Time-series data is typically too constrictive as it contains an exact value for each variable at each time point. Also, measured data is often subject to noise. BioPSy utilises η to relax the time-series data and create an interval of acceptable states for each time point, and returns:

- a set of *feasible* (acceptable) parameter ranges: for all the points in this set, the model is *formally* and *numerically* guaranteed to satisfy the noisy time-series data;
- a set of *infeasible* (unsuitable) parameter ranges: for no point in this set, the model satisfies the noisy time-series data. Again, this is *formally* and *numerically* guaranteed; and

– a set of parameter ranges where, because of the given precision, `BioPSy` is unable to determine if they satisfy the noisy time-series data.

`BioPSy` works by extracting ODEs from the given SBML model along with the list of model parameters. The user can select which parameters ($\overline{p}$) to synthesise and provide initial parameter ranges to search through. `BioPSy` converts the ODEs, parameters, and the noisy time-series data into a collection of SMT problems. Each problem represents an *initial value problem* (IVP) constrained by the initial time point and one of the subsequent time points. Informally, the individual SMT problems contain assertions declaring that the values of each variable (*i.e.*, ODE solution) should be in the interval found in the noisy time-series data after integrating the ODEs for the amount of time between the initial time point and the time point being processed for the file. (Note that for every time point we solve an IVP, and therefore, the first value of the time-series data should not be noisy.) Assertions constraining the parameters being synthesised to be within the synthesised ranges from the previous time point are also added. These constraints help reduce the search space.

The initial boxes for the parameter set are passed one-by-one to the parameter synthesis algorithm, which generates appropriate SMT problems and calls the SMT solver `dReal` [12] to evaluate them. Basically, the synthesis algorithm iteratively splits each box until the minimum size, ϵ, is reached or the current box is either **unsat** or **sat**. (A box needing to be split is denoted **undet**.) An **unsat** outcome means that for no value in the box, the model reaches an acceptable state. A **sat** outcome means that all the values in the box lead the system to an acceptable state. An **undet** outcome means that the algorithm could not decide between **unsat** and **sat**. This indecision might be because the box contains both **sat** and **unsat** regions, or because of the precision, δ, used when solving the SMT problems. This process continues incrementally until all the points in the time series are processed. A high-level workflow for the `BioPSy` tool is presented in Figure 1.

As mentioned, `BioPSy` returns three sets of synthesised parameter ranges, corresponding to **sat**, **unsat**, and **undet**. The **sat** and **unsat** parameter ranges are *formally* and *numerically* guaranteed to be correct. Essentially, these guarantees are made possible by `dReal`, which is based on validated ODE integration and rigorous constraint processing via interval arithmetics.

We note that the precision value, δ, for `dReal` can be arbitrarily small. However, it cannot be zero since solving first-order real formulae with general nonlinear functions is an undecidable problem. (For more information on the theory behind `dReal`, please refer to [11].) Additionally, the precision value, ϵ, can be arbitrarily small. This value determines the level of granularity that the parameter synthesis algorithm uses to search the parameter space. Smaller values mean that `BioPSy` will try to break parameter ranges into smaller segments when searching for acceptable values. The noise parameter, η, controls the size of the intervals produced from the time-series data. Choosing a small η makes it more difficult to identify acceptable ranges, but it produces parameter enclosures that result in the system having better compliance with the original time-series data.

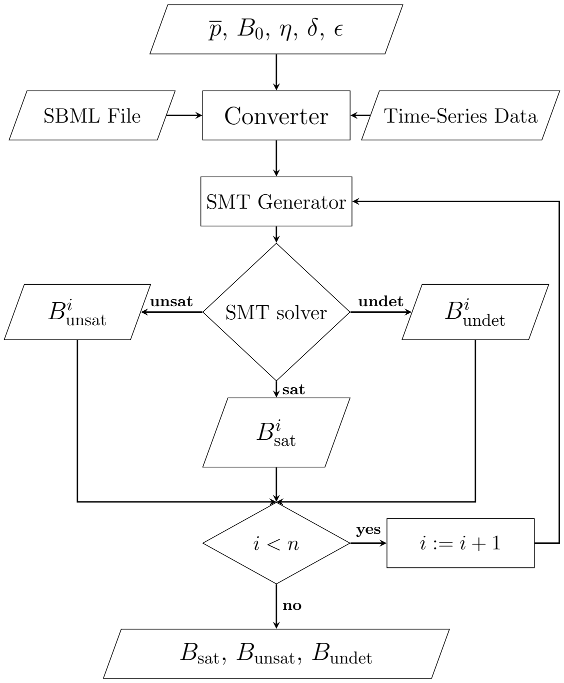

Fig. 1: Workflow diagram for BioPSy. *Legend:* $\overline{p} = \{p_1, \dots p_m\}$: model parameters to synthesise, B_0 = initial set of parameter ranges, η = acceptable noise, δ = SMT solver precision, ϵ = precision of parameter synthesis, n = number of points in the time-series, B^i_{unsat}, B^i_{undet}, and B^i_{sat} = parameter sets for the i-th time point containing boxes for which synthesis is not feasible, undetermined, and feasible, respectively.

Finally, we remark that the main advantage of solving the parameter synthesis problem in a *point-by-point* manner (as we do) is that it reduces the computational complexity, since fewer variables are passed to the SMT solver.

Implementation and Usage The BioPSy *graphical user interface* (GUI) is implemented in Java. The parameter synthesis algorithm is implemented in C++, and it utilises the CAPD library[1] for interval arithmetics and dReal[12] as a standalone application. The algorithm is additionally parallelised using OpenMP. The GUI is launched using the BioPSy JAR file (Java JRE 1.6 or higher required). The user can browse for a model file and a time-series data file. Once selected, these files are shown in the SBML and Time-Series tabs, respectively. The files are also parsed, and the data is displayed in the Parameters and Variables tabs. Under the Parameters tab, the user can select which parameters are to be synthesised, and their precision (ϵ). For synthesised parameters, the user is also able to define a lower bound and upper bound that is used to constrain the

[1] http://capd.ii.uj.edu.pl

parameter search space. Similarly, the Variables tab allows the user to specify bounds on the acceptable values and noise (η) for each variable in the model. Once the bounds are set, clicking the Run button will perform the synthesis. The Advanced Options button enables the user to specify the path to the `dReal` binary as well as the desired level of precision, δ, used by `dReal` ($\delta = 0.001$ is the default value). Once the synthesis has started, the Output tab displays the output file as it is being produced allowing a user to watch as the infeasible ranges, feasible ranges, and undetermined ranges are generated for each time point in the data. The Plot tab displays an updating in real-time graphical representation of the contents of the Output tab (for two parameters only). `BioPSy`'s source code, binary, and the models used in the experiments are available at https://github.com/dreal/biology.

3 Results

`BioPSy` has successfully been applied to several biological models including a model of prostate cancer treatment [15, 20], a model on human starvation [24], and a cell cycle model [25]. In each experiment, two parameters are selected for synthesis while the rest are fixed to the values found in the SBML file. Additionally, the experiments are performed on a 32-core (2.9GHz) Ubuntu Linux machine. The models analysed and their parameters are available at https://github.com/dreal/biology/tree/master/models/CMSB2015.

3.1 Personalized Prostate Cancer Treatment

This model tracks the level of *prostate specific antigen* (PSA) (v) with comprises of two types of cancer cells: *hormone sensitive cells* (HSCs) (x) and *castration resistant cells* (CRCs) (y). In this treatment model, a patient is deprived of androgen (z) causing HRC survival rates to decline. However, lower androgen levels cause HRCs to convert to CRCs and increase the proliferation rate of CRCs. Administrators of this treatment must, therefore, alternate patients between phases of being 'on' and 'off' the treatment in order to prevent both the HSC and the CRC levels from getting out of hand. The ODEs [15, 20] describing the dynamics of a patient on the treatment are shown in Equation (1).

$$\begin{aligned}
\frac{dv}{dt} =& \left(\frac{\alpha_x}{1+\mathrm{e}^{(k_1-z)k_2}} - \frac{\beta_x}{1+\mathrm{e}^{(z-k_3)k_4}} - m_1\left(1-\frac{z}{z_0}\right) - c_1\right)x + c_2 + \\
& m_1\left(1-\frac{z}{z_0}\right)x + \left(\alpha_y\left(1-d_0\frac{z}{z_0}\right)-\beta_y\right)y \\
\frac{dx}{dt} =& \left(\frac{\alpha_x}{1+\mathrm{e}^{(k_1-z)k_2}} - \frac{\beta_x}{1+\mathrm{e}^{(z-k_3)k_4}} - m_1\left(1-\frac{z}{z_0}\right) - c_1\right)x + c_2 \qquad (1) \\
\frac{dy}{dt} =& m_1\left(1-\frac{z}{z_0}\right)x + \left(\alpha_y\left(1-d_0\frac{z}{z_0}\right)-\beta_y\right)y \\
\frac{dz}{dt} =& -z\gamma - c_3
\end{aligned}$$

In this case study, we investigate two applications of BioPSy:

1. **Parameter Synthesis**: given an initial parameter domain and time-series data, we synthesise the parameter sets for which the model is guaranteed to satisfy the time-series; and
2. **Parameter Checking**: we check whether parameter estimates obtained by other methods actually satisfy the time-series.

The first application is, in general, very computationally intensive — its worst-case time complexity grows exponentially with the number of parameters to synthesise. The second application is lighter and gives the user the ability to check if a given parameter value satisfies a desired behaviour of the system.

Parameter Synthesis We perform parameter synthesis using real clinical data [6][2] of a patient who was on treatment for 5 nonconsecutive times throughout 6 years (for about 9 months in each period). The patient was monitored every month and some of the observations (such as PSA and androgen levels) were documented. Overall, every period of time-series data contains around 4-5 time points. For each time-series, we synthesise the parameter set that satisfies the patient's clinical data with noise $\eta = 1.4$ ($\epsilon = 10^{-3}$ and $\delta = 10^{-3}$). The synthesised parameters, α_y and β_x, are explored on the set $[0.0, 0.05] \times [0.0, 0.05]$. The resulting parameter set satisfying all time-series is constructed as the intersection of parameter sets synthesised for each time-series. The feasible set including the ranges $[0.0225, 0.025] \times [0.0325, 0.0332031]$ and $[0.0210938, 0.0225] \times [0.0325, 0.0327344]$. Each time-series evaluation on the specified range took about 12 hours of CPU time. The parameter sets synthesised for each time-series are presented in Figure 2 and the resulting set intersection is shown in Figure 3. We remark that the values for η and ϵ used in our experiments have been chosen purely for didactic reasons. The user can choose more appropriate values depending on the model being studied.

Parameter Checking For this application, parameter values are obtained using the different parameter estimation methods available in COPASI, and the results are verified using BioPSy. These parameter estimation methods utilise simulation-based techniques to explore the parameter space and find a vector of parameters that cause the model to best approximate some time-series data. One downside to these methods is that they are not always capable of finding a satisfying vector of parameters due to the trial-and-error approaches they employ. For the prostate cancer treatment model, every parameter estimation method in COPASI is run using default parameters, and each result takes around 5 seconds to obtain. Some of the methods fail to produce results, but for those methods that are able to find parameters, the estimated values are checked and verified in BioPSy using the same time-series data and $\eta = 1.4$. Each verification took about 10-20s of CPU time, depending on the time-series length. The verification results are presented in Table 1. Here, it can be seen that some of the methods

[2] Data available at: http://www.nicholasbruchovsky.com/clinicalResearch.html

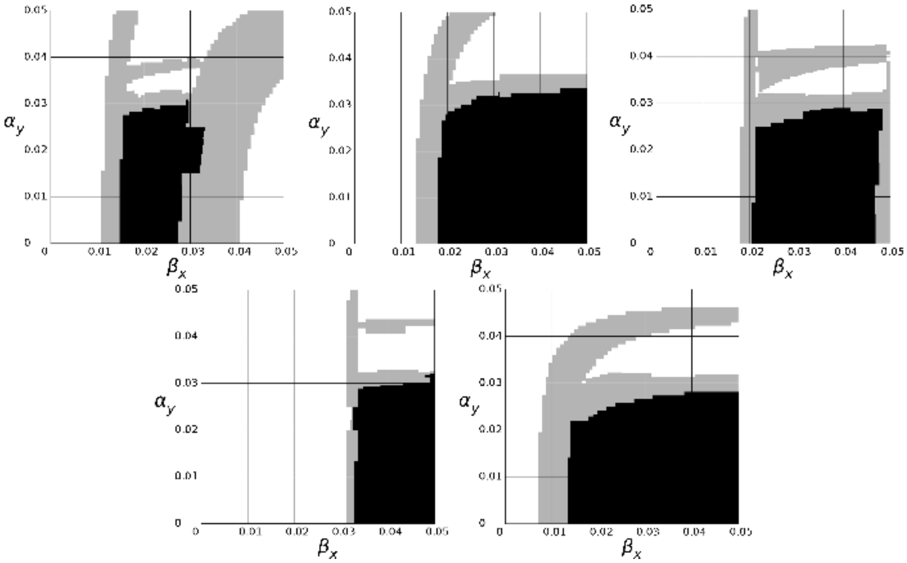

Fig. 2: Prostate cancer model: Parameter synthesis results for β_x and α_y for five time-series (ordered *clockwise*) obtained for each 'on' treatment stage. *Legend*: ***white*** - infeasible boxes; ***black*** - feasible boxes; and ***gray*** - undetermined boxes.

produce results that only satisfy a few of the time-series data, and only one method (Hooke & Jeeves) satisfies all of the data with its parameter values. In contrast, three of the methods return parameters that do not formally satisfy any of the time-series with a noise value equal to 1.4.

3.2 Human Starvation

The human starvation model [24] tracks the amount of fat (F), protein in muscle mass (M), and ketone bodies (K) in the human body after glucose reserves have been depleted from three to four days of fasting. These three variables are modelled using material and energy balances to ensure that the behaviour of the model tracks what is observed in actual experiments involving fasting. The ODEs for this model are presented in Equation (2).

$$\begin{aligned}\frac{dF}{dt} &= F\left(\frac{-a}{1+K} - \frac{1}{\lambda_F}\left(\frac{C+gL_0}{F+M} + \kappa\right)\right)\\ \frac{dM}{dt} &= -\frac{M}{\lambda_M}\left(\frac{C+\kappa L_0}{F+M} + \kappa\right)\\ \frac{dK}{dt} &= \frac{VaF}{1+K} - b\end{aligned} \tag{2}$$

Two parameters are synthesised, κ and b, on the ranges $[9, 11]$ and $[0.05, 0.08]$, respectively, using simulated time-series data that includes 25 time points, $\eta = 0.1$,

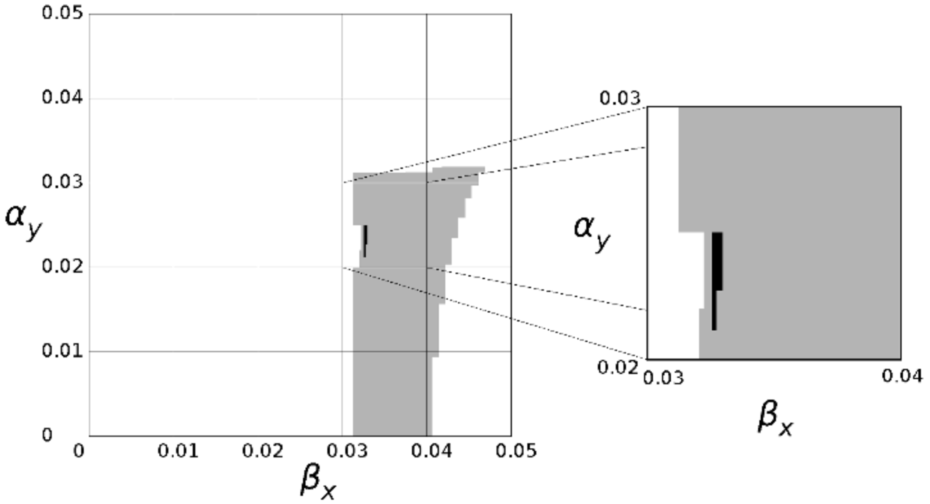

Fig. 3: Prostate cancer model: Parameter synthesis results for β_x and α_y over five time-series obtained for each 'on' treatment stage. *Legend*: ***white*** - infeasible boxes; ***black*** - feasible boxes; and ***gray*** - undetermined boxes.

Table 1: Parameter Checking of `COPASI` results over five time-series on cancer model. *Legend*: n = parameters found invalid by `BioPSy`; y = parameters found valid by `BioPSy`

Method	α_x	α_y	β_x	β_y	`BioPSy`				
					S_1	S_2	S_3	S_4	S_5
Evolut. Prog.	-0.215799	-2.67586×10^{-6}	0.0271774	0.000135248	n	y	y	n	y
Hooke & Jeeves	-0.308608	-0.278566	0.029312	-0.24288	y	y	y	y	y
Levenberg-Marquardt	-0.17045	-31.9428	0.00661261	-10.5429	n	n	n	n	n
Praxis	-0.233483	-0.00697965	0.0240299	0.186801	y	y	y	n	y
Scatter Search	-0.17045	-31.9428	0.00661261	-10.5429	n	n	n	n	n
Simulated Annealing	-0.248778	6.3856×10^{149}	0.0226673	-2.27061×10^{148}	n	n	n	n	n
Truncated Newton	-0.236403	-0.00791949	0.0243545	0.0116282	y	y	y	n	y

$\delta = 0.001$, and $\epsilon = 0.1$. `BioPSy` took 5 minutes and 7 feasible ranges were obtained: $[9.88077, 9.8832] \times [0.0764844, 0.0771875]$, $[9.92213, 10] \times [0.0785938, 0.08]$, $[10, 10.0791] \times [0.0726172, 0.0744629]$, $[9.9416, 10] \times [0.0712109, 0.0761328]$, $[9.8832, 10] \times [0.0761328, 0.0785938]$, $[10, 10.1187] \times [0.0744629, 0.08]$, and $[9.88198, 9.8832] \times [0.0750781, 0.0757813]$. A graphical representation of the final result is shown in Figure 4.

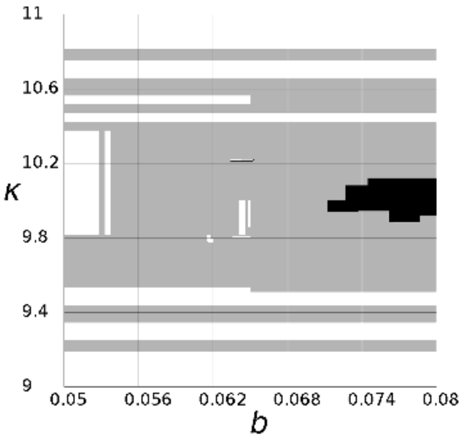

Fig. 4: Human starvation model: Parameter synthesis results for κ and b. *Legend*: ***white*** - infeasible boxes; ***black*** - feasible boxes; and ***gray*** - undetermined boxes.

3.3 Cell Cycle

In the cell cycle model [25], two proteins, CDC2 (u) and Cyclin (v), combine to form a heterodimer that controls major events in a cell causing it to reach a steady state, act as a spontaneous oscillator, or act as an excitable switch. The ODEs for this model are presented in Equation (3).

$$\begin{aligned}\frac{du}{dt} &= k_4\left(v-u\right)\left(\frac{k'_4}{k_4}+u^2\right)-k_6u\\ \frac{dv}{dt} &= \kappa - k_6u\end{aligned} \tag{3}$$

The cell cycle model used has reference BIOMD0000000006 in the BioModels Database [18]. In this example, two parameters are synthesised, k'_4 and k_4, on the ranges $[0.01, 0.02]$ and $[175, 185]$, respectively, using $\eta = 0.001$, $\delta = 0.001$, $\epsilon = 0.1$, from 10 simulated data points. `BioPSy` took 10 minutes to find one feasible range, $[0.0166691, 0.0192934] \times [175, 185]$. This result is shown in Figure 5.

4 Conclusions and Future Work

Here, we present `BioPSy`, an open-source tool for guaranteed parameter set synthesis on biological models from time-series data. `BioPSy` accepts SBML models, so it can be applied to a large number of existing biological models. Indeed, `BioPSy` is not only limited to biological models with mass action kinetics but can handle models involving general ODEs. An important feature about our tool is that models using parameters synthesised with `BioPSy` are formally guaranteed to behave as desired. Also, `BioPSy` can formally validate parameter estimates generated by other methods. We apply `BioPSy` to non-trivial biological models, including a highly nonlinear model of prostate cancer treatment. For this model

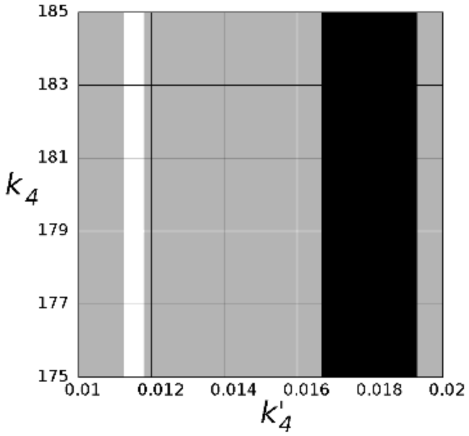

Fig. 5: Cell cycle model: Parameter synthesis results for k_4 and k'_4. *Legend*: ***white*** - infeasible boxes; ***black*** - feasible boxes; and ***gray*** - undetermined boxes.

in particular, BioPSy is able to synthesise parameters from real clinical data. Despite the complexity of parameter synthesis and of the models involved, BioPSy performs reasonably well, and it is usable in practice. We believe BioPSy can be useful for design space exploration in both synthetic and systems biology. In the future, we plan to extend BioPSy to handle biological models that contain both continuous and discrete dynamics — so called *hybrid* models.

Acknowledgements C. M. has been supported by the Engineering and Physical Sciences Research Council (UK) grant EP/K039083/1; F. S. has been supported by award N00014-13-1-0090 of the US Office of Naval Research.

References

1. Arkin, A.: Setting the standard in synthetic biology. Nature Biotech. 26, 771–774 (2008)
2. Asarin, E., Dang, T., Frehse, G., Girard, A., Guernic, C.L., Maler, O.: Recent progress in continuous and hybrid reachability analysis. In: IEEE Conference on Computer Aided Control System Design. pp. 1582–1587 (2006)
3. Balsa-Canto, E., Banga, J.R.: AMIGO, a toolbox for advanced model identification in systems biology using global optimization. Bioinformatics 27(16), 2311–2313 (2011)
4. Barnat, J., Brim, L., Krejci, A., Streck, A., Safranek, D., Vejnar, M., Vejpustek, T.: On parameter synthesis by parallel model checking. IEEE/ACM Trans. Comput. Biol. Bioinformatics 9(3), 693–705 (May 2012)
5. Batt, G., Yordanov, B., Weiss, R., Belta, C.: Robustness analysis and tuning of synthetic gene networks. Bioinformatics 23(18), 2415–2422 (2007)
6. Bruchovsky, N., Klotz, L., et al.: Final results of the Canadian prospective phase II trial of intermittent androgen suppression for men in biochemical recurrence after radiotherapy for locally advanced prostate cancer. Cancer 107, 389–395 (2006)
7. Cimatti, A., Griggio, A., Mover, S., Tonetta, S.: Parameter synthesis with IC3. In: FMCAD. pp. 165–168. IEEE (2013)

8. Donzé, A., Clermont, G., Langmead, C.J.: Parameter synthesis in nonlinear dynamical systems: Application to systems biology. Journal of Computational Biology 17(3), 325–336 (2010)
9. Dötschel, T., Auer, E., Rauh, A., Aschemann, H.: Thermal behavior of high-temperature fuel cells: Reliable parameter identification and interval-based sliding mode control. Soft Comput. 17(8), 1329–1343 (Aug 2013)
10. Dreossi, T., Dang, T.: Parameter synthesis for polynomial biological models. In: HSCC 2014. pp. 233–242. ACM (2014)
11. Gao, S., Avigad, J., Clarke, E.M.: Delta-decidability over the reals. In: LICS. pp. 305–314 (2012)
12. Gao, S., Kong, S., Clarke, E.M.: dReal: An SMT solver for nonlinear theories over the reals. In: CADE. pp. 208–214 (2013)
13. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., Kummer, U.: COPASI - A Complex PAthway SImulator. Bioinformatics 22(24), 3067–3074 (2006)
14. Hucka, M., Finney, A., Sauro, H.M., et al.: The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4), 524–531 (2003)
15. Ideta, A.M., Tanaka, G., Takeuchi, T., Aihara, K.: A mathematical model of intermittent androgen suppression for prostate cancer. Journal of Nonlinear Science 18(6), 593–614 (2008)
16. Jha, S.K., Langmead, C.J.: Synthesis and infeasibility analysis for stochastic models of biochemical systems using statistical model checking and abstraction refinement. Theor. Comput. Sci. 412(21), 2162–2187 (2011)
17. Kitano, H.: Foundations of Systems Biology. MIT Press (Oct 2001)
18. Li, C., et al.: BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Systems Biology 4, 92 (Jun 2010)
19. Little, M.P., Heidenreich, W.F., Li, G.: Parameter identifiability and redundancy: Theoretical considerations. PLoS ONE 5(1), e8915 (01 2010)
20. Liu, B., Kong, S., Gao, S., Zuliani, P., Clarke, E.M.: Towards personalized cancer therapy using delta-reachability analysis. In: HSCC. pp. 227–232. ACM (2015)
21. Maiwald, T., Timmer, J.: Dynamical modeling and multi-experiment fitting with PottersWheel. Bioinformatics 24(18), 2037–2043 (2008)
22. Meslem, N., Ramdani, N., Candau, Y.: Guaranteed parameter set estimation with monotone dynamical systems using hybrid automata. Reliable Computing 14, 88–104 (Jun 2010)
23. Schmidt, H., Jirstrand, M.: Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. Bioinformatics 22(4), 514–515 (2006)
24. Song, B., Thomas, D.: Dynamics of starvation in humans. Journal of Mathematical Biology 54(1), 27–43 (2007)
25. Tyson, J.J.: Modeling the cell division cycle: cdc2 and cyclin interactions. Proceedings of the National Academy of Sciences 88(16), 7328–7332 (1991)
26. Češka, M., Dannenberg, F., Kwiatkowska, M., Paoletti, N.: Precise parameter synthesis for stochastic biochemical systems. In: CMSB 2014. LNCS, vol. 8859, pp. 86–98 (2014)
27. Yordanov, B., Belta, C.: Parameter synthesis for piecewise affine systems from temporal logic specifications. In: HSCC 2008. LNCS, vol. 4981, pp. 542–555 (2008)
28. Zi, Z., Klipp, E.: SBML-PET: a Systems Biology Markup Language-based parameter estimation tool. Bioinformatics 22(21), 2704–2705 (2006)