doi:10.1093/nar/gkt459

SPEDRE: a web server for estimating rate parameters for cell signaling dynamics in data-rich environments

Tri Hieu Nim[1,2], Jacob K. White[1,3] and Lisa Tucker-Kellogg[1,2,4,*]

[1]Computational Systems Biology Programme, Singapore-MIT Alliance, National University of Singapore, 117576, [2]Mechanobiology Institute, National University of Singapore, 117411, Singapore, [3]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA and [4]Department of Dermatology, State University of New York at Stony Brook, NY 11794, USA

Received February 25, 2013; Revised April 27, 2013; Accepted May 6, 2013

ABSTRACT

Cell signaling pathways and metabolic networks are often modeled using ordinary differential equations (ODEs) to represent the production/consumption of molecular species over time. Regardless whether a model is built *de novo* or adapted from previous models, there is a need to estimate kinetic rate constants based on time-series experimental measurements of molecular abundance. For data-rich cases such as proteomic measurements of all species, spline-based parameter estimation algorithms have been developed to avoid solving all the ODEs explicitly. We report the development of a web server for a spline-based method. Systematic Parameter Estimation for Data-Rich Environments (SPEDRE) estimates reaction rates for biochemical networks. As input, it takes the connectivity of the network and the concentrations of the molecular species at discrete time points. SPEDRE is intended for large sparse networks, such as signaling cascades with many proteins but few reactions per protein. If data are available for all species in the network, it provides global coverage of the parameter space, at low resolution and with approximate accuracy. The output is an optimized value for each reaction rate parameter, accompanied by a range and bin plot. SPEDRE uses tools from COPASI for pre-processing and post-processing. SPEDRE is a free service at http://LTKLab.org/SPEDRE.

INTRODUCTION

Mathematical modeling of biochemical network dynamics using ordinary differential equations (ODEs) has yielded impressive advances in our understanding of complex biological systems (1). When constructing an ODE model, there is often a need to estimate kinetic rate constants based on time-series experimental measurements of molecular concentrations or enzyme activities. Even when time-series experimental measurements of all species are available [such as by using stable isotope labeling by amino acids in cell culture (SILAC) proteomics (2)], estimating the rate constants is still a difficult non-linear optimization problem (3). Many widely used methods, collected into popular software packages such as COmplex PAthway SImulator (COPASI) (4), are applicable to this parameter estimation problem. Methods have traditionally been classified as local, global or hybrid global + local methods (5,6).

The traditional application of parameter estimation for modeling network dynamics has been to small biochemical networks with sparse data sets. With the growing ease of measuring complete proteomes (7) and with the assembly of large network models, needs have expanded to include to data-rich approaches to parameter estimation, sometimes called spline-based collocation methods (5,8–10). Spline-based collocation methods exploit complete or nearly complete data sets to interpolate directly the slopes of concentration over time, instead of relying on numerical simulations to compute the derivatives of the ODEs. Spline-based collocation methods have not previously been implemented in any parameter estimation web server. The method of Systematic Parameter Estimation for Data-Rich Environments (SPEDRE) (11) uses a spline-based collocation approach and coarse-grained discretization to provide approximate heuristic search of the global parameter space, with excellent scalability for large networks. However, SPEDRE is designed only for problems with low-degree networks (few reactions per protein, but no limit on the number of proteins). Although there are many algorithms for parameter estimation [reviewed in (12,13)], we are not aware of

*To whom correspondence should be addressed. Tel: +1 415 675 1968; Fax: +65 6872 6123; Email: LisaTK@nus.edu.sg

any parameter estimation web servers, other than COPASI (14,15) and SPEDRE.

PROCESSING METHOD

Parameter estimation in biological networks is challenging because the parameters are interdependent, which makes them impossible to estimate individually. Even if each parameter can take on only a fixed number of possible values, the possible combinations of these parameter values becomes astronomical, growing exponentially with respect to the number of reactions. SPEDRE exploits complete data sets to interpolate the slopes (concentration change over time) instead of simulating the whole system, and it exploits the low degree of the network to construct a linear number of sparsely connected low-dimensional subproblems.

The pipeline for SPEDRE, illustrated in Figure 1, consists of five stages: input, pre-computing discretized tables for each subproblem, Loopy Belief Propagation to merge the subproblems, local optimization to refine the coarse-grained solution and output. During the input stage, SPEDRE requires the user to provide the network connectivity, the reaction types (Michaelis–Menten, mass action kinetics, etc.), the time-series measurements and some optional runtime settings. In the discretization stage, SPEDRE transforms the continuous range of rate constants into discrete bins, and it pre-computes lookup tables with discretized solutions to each ODE. The next stage must construct a single system-wide parameter vector by looking up and merging the best parameter combinations from the low-dimensional subproblems. We do this heuristically using Loopy Belief Propagation (16), also called 'message passing', a probabilistic network inference technique that computes probability distributions for the parameters and sends the distributions as messages across the edges of the network. Loopy Belief Propagation is an iterative heuristic that terminates by convergence or when the specified maximum number of iterations is reached. On termination, Loopy Belief Propagation provides optimized bins for all rate constants. This set of bins provides a starting point for the post-processor to refine, using the Levenberg–Marquardt numerical method of local optimization (17). In essence, SPEDRE is a hybrid global + local optimization method, but unlike other hybrid global + local methods, the global portion (called SPEDRE-base) does not use stochastic sampling. The final output of SPEDRE is a plot of the bins provided by Loopy Belief Propagation, as well as a vector of the optimized rate parameters. Our previous work specified the SPEDRE algorithm in detail (11), whereas the current work aims to describe the web server interface.

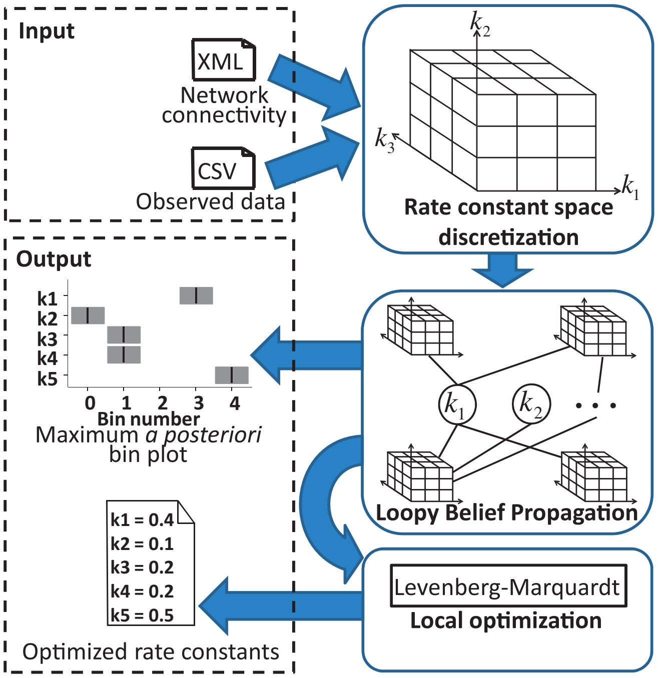


Figure 1. Pipeline of the processing methods underlying the SPEDRE web server.

Asymptotic analysis of the underlying SPEDRE-base algorithm (11) reveals attractive properties of the method because the time complexity scales exponentially with the network degree, but it scales efficiently (polynomially) with the number of species, the number of reactions and the size of the data set. Correspondingly, the method scales well on biological pathways with a bounded number of reactions per species. Dense networks with hub-like species have high network degree and are unsuitable for the SPEDRE algorithm. Our web server handles these cases simply by running the Levenberg–Marquardt algorithm (17) instead.

The SPEDRE-base algorithm was implemented in C++ with an interface to COPASI (version 4.6, build 32) for the Levenberg–Marquardt algorithm. The web server version of SPEDRE was implemented using the Opal toolkit, as introduced in (18). To display the customizable bin plot of SPEDRE results, Google Chart API (Google Inc.) was used. As the features of Opal and Google Charts API grow over time, the functionality of the SPEDRE web server will grow as well.

INPUT

SPEDRE requires two main inputs from users: the concentrations of all the molecular species and the biochemical reactions. The concentrations of the molecular species would come from proteomic experiments, from computational hypotheses that merge published sub-networks or from any source that specifies the abundance of every species, at a set of discrete timepoints. This requirement for input of all species is highly restrictive, but it allows SPEDRE to focus on data-rich problems, which are a specialized but growing segment of parameter estimation work. The concentrations must be specified in a comma separated value (CSV or TXT) file. The file format is compatible with time-course simulation output from the COPASI software, where the first line specifies the headers (time and species names). The first column of each additional row specifies the time value, with the species levels in the remaining columns. If a data point is found to be invalid, the relevant time point is removed from the system before performing rate constant estimation. If there is greater incompleteness or sparsity in the available measurements, users should use the CopasiWeb (14) service instead.

The biochemical reactions must be specified as an XML file in COPASI_ML format (4), which can be obtained from SBML format using a link to the conversion service in CopasiWeb (14). This format includes 'rate laws' with predefined reaction types, which are necessary for constructing the correct types of kinetic parameters. SPEDRE is currently restricted to the following reaction types: 'Michaelis-Menten catalysis', 'Mass action (irreversible)' and 'Enzyme simple' [rate_constant × (enzyme) × (substrate)]. In other words, a reversible reaction must be represented using two separate reactions, one in each direction; reactions with high-order combinations must be re-expressed as a series of subreactions. Future work may automate the conversion process. The SMBL conversion service provided at CopasiWeb (14), may disrupt the names of the reaction types ('rate laws'), in which case users must change the names of the reaction types manually.

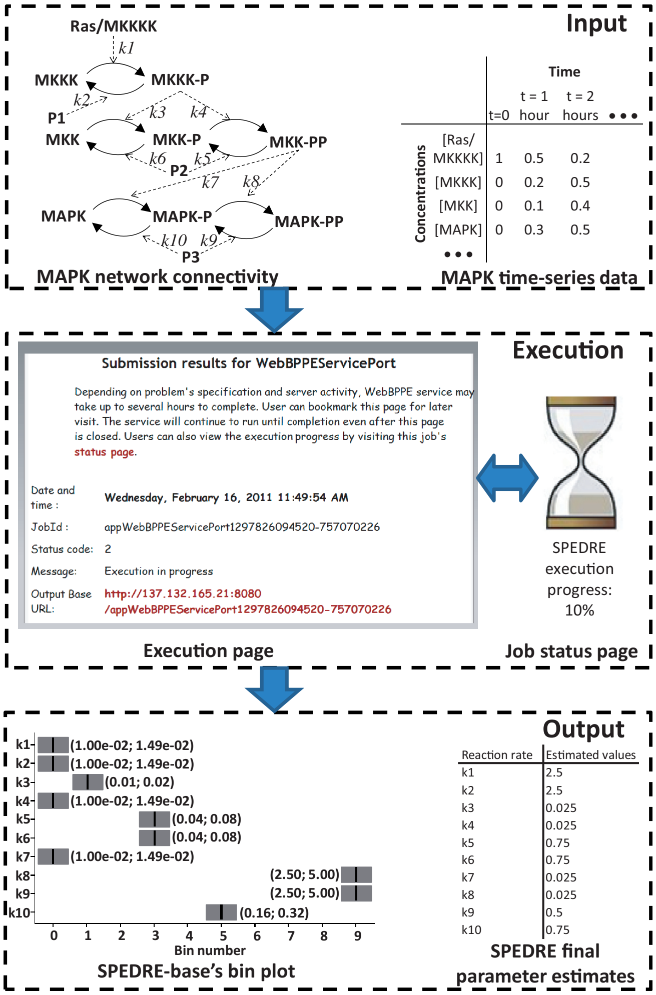
The web server homepage provides descriptions and illustrations for several published pathways, including the Akt pathway (19), the MAPK pathway (20) and a pathway of Actin Filament Assembly-Disassembly (21). Also available is a spectrum of artificial networks (circular or tree-shaped) with widely varying sizes.

A set of default parameters can be modified using the submission form, if users wish to adjust how SPEDRE executes. The main options are the number of bins for discretizing the parameter ranges, and the number of iterations for Loopy Belief Propagation. In addition, the bin spacing can be set to linear or logarithmic scaling. The upper and lower bounds of the rate constants can be specified globally, or individual rate constants can override the upper bound and lower bound if these are specified in the network connectivity input file. The maximum number of iterations can be set to zero if users wish to perform a standalone local search. The anticipated error rate is an option for theoretical calculations, and it allows users to add Gaussian noise to the observed data. Another option for specialized users, called 'samples per voxel', allows each voxel of parameter space (each set of parameter bins) to be evaluated by sampling multiple random points in the voxel, instead of using the voxel midpoint.

OUTPUT

An example execution based on the MAPK cascade is shown in Figure 2. Using the web interface, users can submit input files that follow the specified formats (Figure 2, top box), and SPEDRE performs the computation task while simultaneously displaying the execution page (Figure 2, middle box).

Different execution specifications may result in different runtime performance, and some jobs may require several hours to complete. Users may wish to bookmark the location of the output page for a later visit. A status page is linked to the execution page and shows the percentage of the overall task that has been completed. Users are advised to consult the asymptotic analysis of SPEDRE-base algorithm (11) when adjusting the SPEDRE execution options beyond the default values. On completion, SPEDRE returns an estimated range for each rate constant, in the form of a bin plot, as shown in Figure 2 (bottom box). A bin plot is a visual representation of the resulting voxel in high-dimensional space, which gives users an impression of the exponential number of possible combinations of rate constants, even in a coarsely discretized search space. Each bin indicates a range in which the estimated rate parameter lies. Finally, the bin midpoints are used as a starting point for local optimization, and the refined set of parameters are output as a vector of floating point numbers.



		Time			
		t=0	t = 1 hour	t = 2 hours	• • •
Concentrations	[Ras/ MKKKK]	1	0.5	0.2	
	[MKKK]	0	0.2	0.5	
	[MKK]	0	0.1	0.4	
	[MAPK]	0	0.3	0.5	
	• • •				

Submission results for WebBPPEServicePort

Depending on problem's specification and server activity, WebBPPE service may take up to several hours to complete. User can bookmark this page for later visit. The service will continue to run until completion even after this page is closed. Users can also view the execution progress by visiting this job's status page.

Date and time :	Wednesday, February 16, 2011 11:49:54 AM
JobId :	appWebBPPEServicePort1297826094520-757070226
Status code:	2
Message:	Execution in progress
Output Base URL:	http://137.132.165.21:8080 /appWebBPPEServicePort1297826094520-757070226

Reaction rate	Estimated values
k1	2.5
k2	2.5
k3	0.025
k4	0.025
k5	0.75
k6	0.75
k7	0.025
k8	0.025
k9	0.5
k10	0.75

Figure 2. SPEDRE execution results using the MAPK network derived from (20). Additional information about this test case is provided on the server website.

The bin plot in Figure 2 was generated using the Google Chart API (Google Inc.), which imposes certain constraints, including the maximum URL length of 2048 characters (for plot formatting) and maximum plot size of 300 000 pixels. Users may encounter cluttered plots

Table 1. Performance of the SPEDRE web service on a series of test cases: synthetic networks (circular network, tree network with random branch) and biological networks (PI3K/Akt cascade, MAPK cascade and Actin Filament Assembly/Disassembly pathway)

Test Case	Weighted SSE (unitless)	SPEDRE-base run-time (S)	Total SPEDRE run-time (S)
1. Circular network (80 species)	0.71	17.00	20.17
2. Random low-degreed network (80 species)	1.56	75.00	79.26
3. P13K/Akt cascade	6.72E-07	25.00	25.41
4. MAPK cascade	9.29E-07	91.00	91.51
5. Actin Filament Assembly/Disassembly	1.37	468.00	468.51

Weighted SSE (objective function value): sum-of-square error, weighted by mean square of each species concentrations across all time points, between simulated and given time series. SPEDRE was executed with lower bound = 0, upper bound = 1, logarithmically spaced binning with five bins, maximum number of iterations = 5, Gaussian noise = 0 and number of samples per voxel = 5.

for large network size (>30 rate constants). As the API is actively developed with large user base across the industry, this current limit will be overcome with new updates of the API.

PERFORMANCE

Depending on execution configuration, SPEDRE can achieve various performance outcomes. Table 1 shows the web server's performance on five test cases using noiseless and 20% noise data. The weighted sum-of-square error (SSE) represents how well a model with estimated parameters could fit the input time-series data.

These test cases used coarse discretization to run quickly at the expense of accuracy. For the PI3K/Akt cascade and MAPK cascade, the objective function was low, indicating good match with data. (A good match with data means the parameterized model gives plausible explanations of the data, but alternative models or parameters may also exist). The runtime of SPEDRE base (i.e. SPEDRE without Levenberg–Marquardt) was close to the total SPEDRE runtime, indicating that the hybrid global-local approach incurs low additional runtime cost compared with global search alone. The runtime measures also provide an empirical demonstration that SPEDRE runtime scales efficiently with the size of the input network and poorly with the degree of the network. Specifically, the high-degreed Actin Filament Assembly/Disassembly pathway has only 14 species and 25 reactions, whereas the low-degreed circular network has 80 species and 80 reactions; yet, execution on the latter network completes ~23 times faster than the former.

DISCUSSION

SPEDRE has been implemented as a web-based service for performing rate constant estimation on biochemical networks, such as cell signaling pathways and metabolic networks. SPEDRE uses a spline-based collocation approach, requiring extensive data as input and providing efficient coverage of enormous parameter spaces. The computational power of a web server makes it suitable for intensive rate constant estimation jobs. The server has dynamic display of the bin plot, as shown in Figure 2 (bottom box), which is customizable using JavaScript. SPEDRE performs preprocessing of user inputs to eliminate missing or invalid data points from the data file. In the scenarios involving a dense network or other features that violate the requirements of SPEDRE, the web service will perform Levenberg–Marquardt optimization only, as an automatic 'rescue' for the parameter estimation problems.

This service is not predictive because measured rate constants are not yet available for pathways of significant size. For users who wish to address the accuracy of parameter estimation as a purely mathematical problem, artificial data sets are available and the weighted SSE is displayed.

FUNDING

Singapore-MIT Alliance IUP [R-154-001-348-646 to L.T.K. and J.K.W.]; Mechanobiology Institute (to L.T.K.). Funding for open access charge: Singapore-MIT Alliance [N-382-000-014-001].

Conflict of interest statement. None declared.

References

1. Fall,C., Marland,E., Wagner,J. and Tyson,J. (2002) *Computational cell biology*. Springer, New York, USA.
2. Mann,M. (2006) Functional and quantitative proteomics using SILAC. *Nat. Rev. Mol. Cell Biol.*, **7**, 952–958.
3. Kleinstein,S.H., Bottino,D., Lett,G.S., Georgieva,A. and Sarangapani,R. (2006) Nonuniform sampling for global optimization of kinetic rate constants in biological pathways. In: *Proceedings of the 2006 Winter Simulation Conference, Vols 1–5*. Institute of Electrical and Electronics Engineers, New York, USA, pp. 1611–1616, 2307.
4. Hoops,S., Sahle,S., Gauges,R., Lee,C., Pahle,J., Simus,N., Singhal,M., Xu,L., Mendes,P. and Kummer,U. (2006) COPASI—a COmplex PAthway SImulator. *Bioinformatics*, **22**, 3067–3074.
5. Chou,I.C. and Voit,E.O. (2009) Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Math. Biosci.*, **219**, 57–83.
6. Moles,C., Mendes,P. and Banga,J. (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.*, **13**, 2467–2474.
7. Mann,M., Kulak,N.A., Nagaraj,N. and Cox,J. (2013) The coming age of complete, accurate, and ubiquitous proteomes. *Mol. Cell*, **49**, 583–590.
8. Zhan,C. and Yeung,L.F. (2011) Parameter estimation in systems biology models using spline approximation. *BMC Syst. Biol.*, **5**, 14.
9. Ramsay,J.O., Hooker,G., Campbell,D. and Cao,J. (2007) Parameter estimation for differential equations a generalized smoothing approach. *J. R. Stat. Soc. B Stat. Method.*, **69**, 741–796.
10. Jia,G., Stephanopoulos,G.N. and Gunawan,R. (2011) Parameter estimation of kinetic models from metabolic profiles: two-phase dynamic decoupling method. *Bioinformatics*, **27**, 1964–1970.
11. Nim,T.H., Luo,L., Clément,M.V., White,J.K. and Tucker-Kellogg,L. (2013) Systematic parameter estimation in data-rich environments for cell signaling dynamics. *Bioinformatics*, **29**, 1044–1051.

12. Moles,C.G., Mendes,P. and Banga,J.R. (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.*, **13**, 2467–2474.
13. Rodriguez-Fernandez,M., Mendes,P. and Banga,J.R. (2006) A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems*, **83**, 248–265.
14. Dada,J. and Mendes,P. (2009) Design and architecture of web services for simulation of biochemical systems. In: Paton,N., Missier,P. and Hedeler,C. (eds), *Lecture Notes in Computer Science*, Vol. 5647. Springer, Berlin/Heidelberg, pp. 182–195.
15. Kent,E., Hoops,S. and Mendes,P. (2012) Condor-COPASI: high-throughput computing for biochemical networks. *BMC Syst. Biol.*, **6**, 91.
16. Murphy,K., Weiss,Y. and Jordan,M. (1999) Loopy belief propagation for approximate inference: an empirical study. In: *Proceedings of Uncertainty in AI,* 467. Morgan Kaufmann, San Francisco, California, USA.
17. Marquardt,D. (1963) An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, **11**, 431.
18. Ren,J., Williams,N., Clementi,L., Krishnan,S. and Li,W.W. (2010) Opal web services for biomedical applications. *Nucleic Acids Res.*, **38**, W724–W731.
19. Hatakeyama,M., Kimura,S., Naka,T., Kawasaki,T., Yumoto,N., Ichikawa,M., Kim,J.H., Saito,K., Saeki,M., Shirouzu,M. *et al.* (2003) A computational model on the modulation of mitogen-activated protein kinase (MAPK) and Akt pathways in heregulin-induced ErbB signalling. *Biochem. J.*, **373**, 451–463.
20. Huang,C.Y. and Ferrell,J.E. (1996) Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc. Natl Acad. Sci. USA*, **93**, 10078–10083.
21. Berro,J., Sirotkin,V. and Pollard,T.D. (2010) Mathematical modeling of endocytic actin patch kinetics in fission yeast: disassembly requires release of actin filament fragments. *Mol. Biol. Cell*, **21**, 2905–2915.