

RESEARCH ARTICLE

PEtab—Interoperable specification of parameter estimation problems in systems biology

Leonard Schmiester^{1,2}, Yannik Schälte^{1,2}, Frank T. Bergmann³, Tacio Camba^{4,5}, Erika Dudkin⁶, Janine Egert^{7,8}, Fabian Fröhlich⁹, Lara Fuhrmann⁶, Adrian L. Hauber^{8,10}, Svenja Kemmer^{8,10}, Polina Lakrisenko^{1,2}, Carolin Loos^{1,2,11,12}, Simon Merkt⁶, Wolfgang Müller¹³, Dilan Pathirana⁶, Elba Raimúndez^{1,2,6}, Lukas Refisch^{7,8}, Marcus Rosenblatt^{8,10}, Paul L. Stapor^{1,2}, Philipp Städter^{1,2}, Dantong Wang^{1,2}, Franz-Georg Wieland^{8,10}, Julio R. Banga⁵, Jens Timmer^{8,10,14}, Alejandro F. Villaverde⁵, Sven Sahle³, Clemens Kreutz^{7,8,14}, Jan Hasenauer^{1,2,6†*}, Daniel Weindl^{1†}



1 Institute of Computational Biology, Helmholtz Zentrum München—German Research Center for Environmental Health, Neuherberg, Germany, **2** Center for Mathematics, Technische Universität München, Garching, Germany, **3** BioQUANT/COS, Heidelberg University, Heidelberg, Germany, **4** Department of Applied Mathematics II, University of Vigo, Vigo, Galicia, Spain, **5** BioProcess Engineering Group, IIM-CSIC, Vigo, Galicia, Spain, **6** Faculty of Mathematics and Natural Sciences, University of Bonn, Bonn, Germany, **7** Faculty of Medicine and Medical Center, Institute of Medical Biometry and Statistics, University of Freiburg, Freiburg, Germany, **8** Freiburg Center for Data Analysis and Modeling (FDM), University of Freiburg, Freiburg, Germany, **9** Department of Systems Biology, Harvard Medical School, Boston, Massachusetts, USA, **10** Institute of Physics, University of Freiburg, Freiburg, Germany, **11** Ragon Institute of MGH, MIT and Harvard, Cambridge, Massachusetts, USA, **12** Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, **13** Heidelberg Institute for Theoretical Studies (HITS gGmbH), Heidelberg, Germany, **14** Signalling Research Centres BIOSS and CIBSS, University of Freiburg, Freiburg, Germany

 These authors contributed equally to this work.
 †JH and DW also contributed equally to this work.
 * jan.hasenauer@uni-bonn.de

 OPEN ACCESS

Citation: Schmiester L, Schälte Y, Bergmann FT, Camba T, Dudkin E, Egert J, et al. (2021) PÉtab—Interoperable specification of parameter estimation problems in systems biology. *PLoS Comput Biol* 17(1): e1008646. <https://doi.org/10.1371/journal.pcbi.1008646>

Editor: Dina Schneidman-Duhovny, Hebrew University of Jerusalem, ISRAEL

Received: August 7, 2020

Accepted: December 18, 2020

Published: January 26, 2021

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: <https://doi.org/10.1371/journal.pcbi.1008646>

Copyright: © 2021 Schmiester et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The authors confirm that all data underlying the findings are fully available without restriction. Specifications of PÉtab, the PÉtab Python library, as well as links to

Abstract

Reproducibility and reusability of the results of data-based modeling studies are essential. Yet, there has been—so far—no broadly supported format for the specification of parameter estimation problems in systems biology. Here, we introduce PÉtab, a format which facilitates the specification of parameter estimation problems using Systems Biology Markup Language (SBML) models and a set of tab-separated value files describing the observation model and experimental data as well as parameters to be estimated. We already implemented PÉtab support into eight well-established model simulation and parameter estimation toolboxes with hundreds of users in total. We provide a Python library for validation and modification of a PÉtab problem and currently 20 example parameter estimation problems based on recent studies.

examples, and all supporting software tools are available at <https://github.com/PEtab-dev/PEtab> a snapshot is available at <https://doi.org/10.5281/zenodo.3732958>. All original content is available under permissive licenses.

Funding: This work was supported by the European Union's Horizon 2020 research and innovation program (CanPathPro; Grant no. 686282; J.H., D.W., P.L.S., E.D., S.M., A.F.V., J.R. B.), the German Federal Ministry of Education and Research (Grant no. 01ZX1916A; D.W., P.L. & 01ZX1705A; J.H., P.L. & Grant no. 031L0159C; J. H. & de.NBI ModSim1, 031L0104A; F.T.B. & EA: Sys, 031L0080; J.E. L.R & Grant no. 031L0048; S. K., F.G.W. & 01ZX1310B; E.R.), the German Federal Ministry of Economic Affairs and Energy (Grant no. 16KN074236; D.P.), the German Research Foundation (Grant no. HA7376/1-1; Y.S.; CIBSS-EXC-2189-2100249960-390939984; C.K., J.T.; project ID: 272983813 - TRR 179; M.R. & Clusters of Excellence EXC 2047 and EXC 2151; E. R., J.H.), the Deutsche Krebshilfe (Grant no. 70112355; A.L.H.), the Human Frontier Science Program (Grant no. LT000259/2019-L1; F.F.), the National Institute of Health (Grant no. U54-CA225088; F.F.). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Author summary

Parameter estimation is a common and crucial task in modeling, as many models depend on unknown parameters which need to be inferred from data. There exist various tools for tasks like model development, model simulation, optimization, or uncertainty analysis, each with different capabilities and strengths. In order to be able to easily combine tools in an interoperable manner, but also to make results accessible and reusable for other researchers, it is valuable to define parameter estimation problems in a standardized form. Here, we introduce P_Etab, a parameter estimation problem definition format which integrates with established systems biology standards for model and data specification. As the novel format is already supported by eight software tools with hundreds of users in total, we expect it to be of great use and impact in the community, both for modeling and algorithm development.

Introduction

Dynamical modeling is central to systems biology, providing insights into the underlying mechanisms of complex phenomena [1]. It enables the integration of heterogeneous data, the testing and generation of hypotheses, and experimental design. However, to achieve this, the unknown model parameters commonly need to be inferred from experimental observations.

Various software tools exist for simulating models and inferring parameters [2–10], which implement various methods and algorithms. Many of these tools support community standards for model specification to facilitate reproducibility, interoperability and reusability. In particular the Systems Biology Markup Language (SBML) [11], CellML [12] and the BioNet-Gen Language (BNGL) [13] are widely used.

The Simulation Experiment Description Markup Language (SED-ML) builds on top of such model definitions and allows for a machine-readable description of simulation experiments based on XML [14]. Also more complex simulation experiments like parameter scans can be encoded, and a human-readable adaptation is provided by the phraSED-ML format [15]. Similarly, the XML-based Systems Biology Results Markup Language (SBRML) was designed to associate models with experimental data and share simulation experiment results in a machine-readable way [16]. Like SED-ML, SBRML can also be used for parameter scans. Complementary, SBtab is a set of table-based conventions for the definition of experimental data and models designed for human-readability and -writability [17].

However, parameter estimation is so far not in the scope of any of the available formats, and important information for it, like the definition of a noise model, is missing. Parameter estimation toolboxes usually use their own specific input formats, making it difficult for the user to switch between tools to benefit from their complementary functionalities and hindering reusability and reproducibility.

Based on our experience with parameter estimation and tool development for systems biology, we developed P_Etab, a tabular format for specifying parameter estimation problems. This includes the specification of biological models, observation and noise models, experimental data and their mapping to the observation model, as well as parameters in an unambiguous way.

Design and implementation

Scope

The scope of PEtab is the full specification of parameter estimation problems in typical systems biology applications. In our experience, a typical setup of data-based modeling starts either with (i) the model of a biological system that is to be calibrated, or with (ii) experimental data that are to be integrated and analyzed using a computational model. Measurements are linked to the biological model by an observation and noise model. Often, measurements are taken after some perturbations have been applied, which are modeled as derivations from a generic model (Fig 1A). Therefore, one goal was to specify such a setup in the least redundant way. Furthermore, we wanted to establish an intuitive, modular, machine- and human-readable and -writable format that makes use of existing standards.

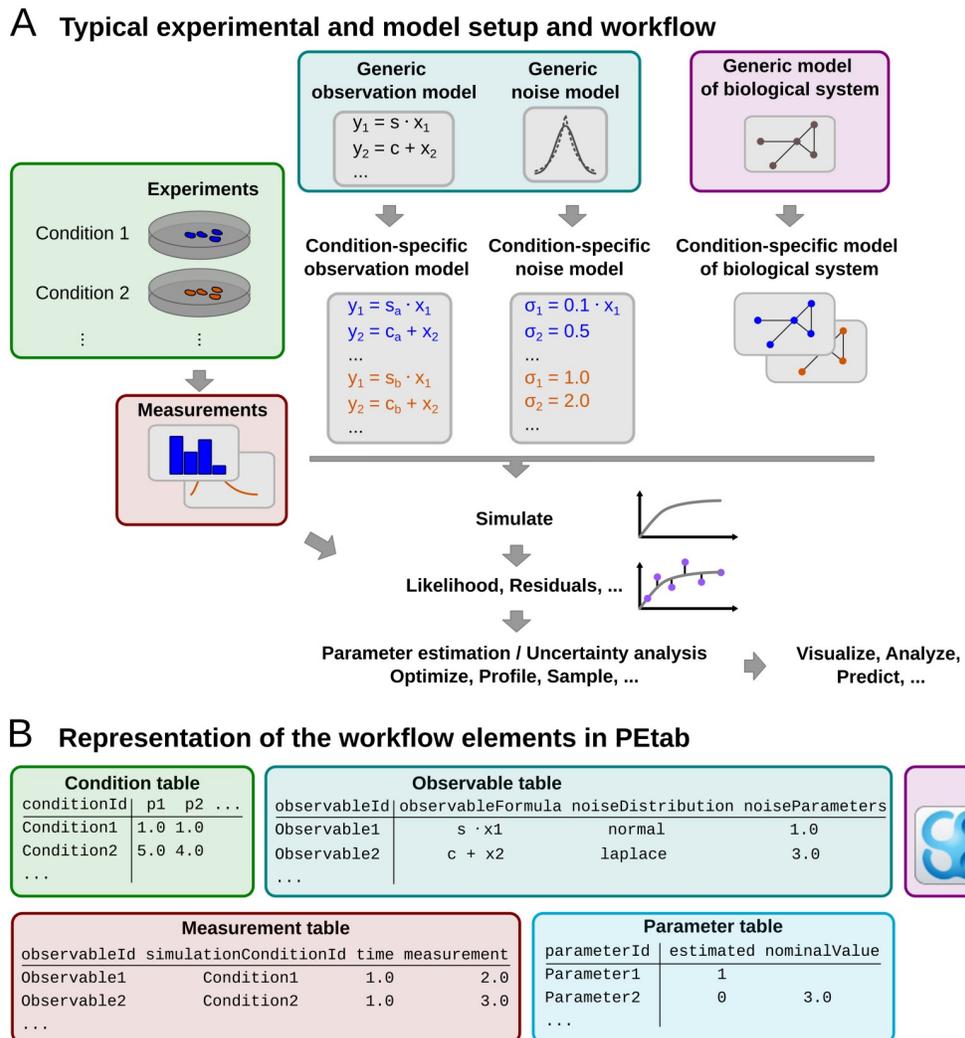


Fig 1. Specifying parameter estimation problems in PEtab. (A) Example of a typical setup for data-based modeling. Usually, a model of a biological system is developed and calibrated based on measurements from perturbation experiments, which are linked to the biological model by an observation model. Different instances of a generic model are used to account for different perturbations or measurement setups. (B) Simplified illustration of how different entities from (A) map to different PEtab files (not all table columns are shown).

<https://doi.org/10.1371/journal.pcbi.1008646.g001>

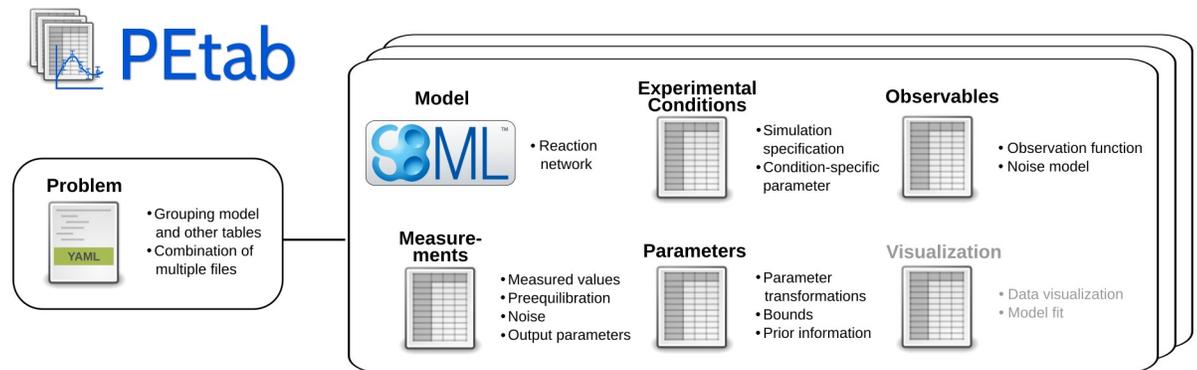


Fig 2. Overview of PEPtab files and the most important features. PEPtab consists of a model in the SBML format and several tab-separated value (TSV) files to specify measurements and link them to the model. A visualization file can be provided optionally. A YAML file can be used to group the aforementioned files unambiguously.

<https://doi.org/10.1371/journal.pcbi.1008646.g002>

PEtab problem specification format

PEtab defines parameter estimation problems using a set of files that are outlined in Fig 2. A detailed specification of PEPtab version 1 is provided in supplementary file [S1 File](#), as well as at <https://github.com/PEtab-dev/PEtab>. Additionally, we created a tutorial illustrating how to set up a PEPtab problem, covering the most common features (supplementary file [S2 File](#)). Further example problems can be found at <https://github.com/Benchmarking-Initiative/Benchmark-Models-PEtab>. The different files specify the biological model, the observation model, experimental conditions, measurements, parameters and visualizations (Fig 1B). These files are described in more detail in the following.

Model (SBML): File specifying the biological process using the established and well-supported SBML format [11]. Any existing SBML model can be used without modification. All versions of SBML are supported by PEPtab and can be used if the specific toolbox supports it.

Experimental conditions (TSV): File specifying the condition(s), such as drug stimuli or genetic backgrounds, under which the experimental data were collected. These experimental conditions specify model properties that are altered between conditions, and allow for a hierarchical specification of model properties (Fig 3A). If simulation conditions are used for pre-equilibration—meaning that some experiment started from the equilibrium reached for another condition—specific model states can be marked for re-initialization (Fig 3B).

Observables (TSV): File linking model properties such as state variables and parameter values to measurement data via observation functions and noise models. Various noise models including normal and Laplace distributions are supported, and noise model parameters can be estimated. Observables can be on linear or logarithmic scale.

Measurements (TSV): File specifying and linking experimental data to the experimental conditions and the observables via the respective identifiers. Optionally, simulation conditions for pre-equilibration can be defined (Fig 3B). Parameters that are relevant for the observation process of a given measurement, such as offsets or scaling parameters, can be provided along with the measured values. This allows for overriding generic output parameters in a measurement-specific manner (Fig 3A).

Parameters (TSV): File defining the parameters to be estimated, including lower and upper bounds as well as transformations (e.g., linear or logarithmic) to be used in parameter estimation. Furthermore, prior information on the parameters can be specified to inform starting points for parameter estimation, or to perform Bayesian inference.

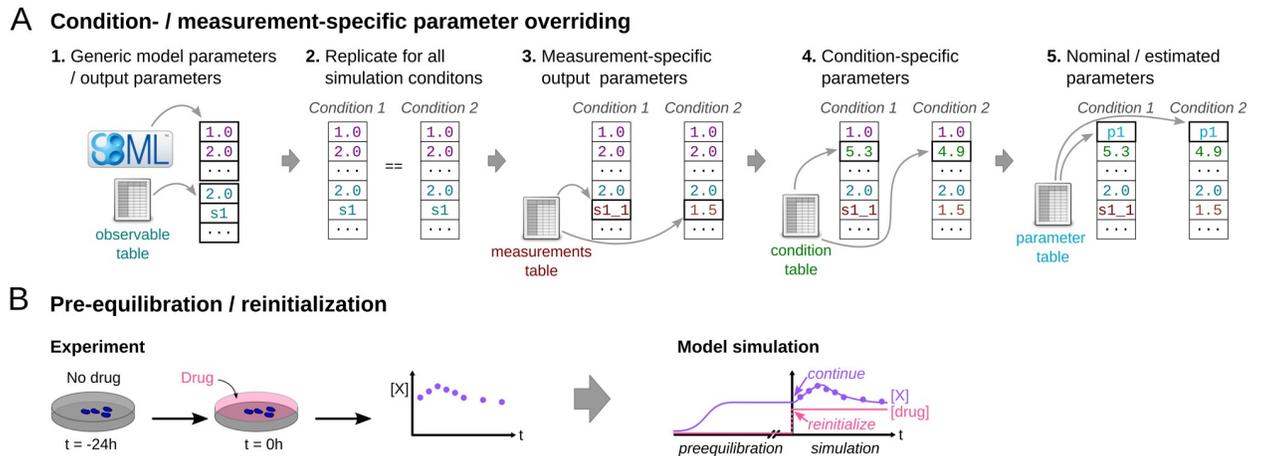


Fig 3. Parameter hierarchy and pre-equilibration in PETA. (A) Illustration of possibilities and precedence of parameter overriding at different stages. The generic model parameter vector, as specified in the SBML model, can be overridden via the observable, measurement, condition and parameter tables, differentially for conditions and measurement points to account for different model inputs or observational model parameters. The parameters that are overridden in each step are indicated with thicker cell borders. Individual parameters can be set to specific values or marked to be estimated (as here p_1). (B) In an often encountered experimental setup, a biological system is under some “baseline” condition and assumed to be in equilibrium (e.g., here depicted for after 24h incubation) before a perturbation is applied. If the equilibrium state of the system is not known a priori, such a setup can be modeled by simulating the system until an apparent steady-state is reached (pre-equilibration). To simulate the perturbation, a subset of model states are reinitialized.

<https://doi.org/10.1371/journal.pcbi.1008646.g003>

Visualization (TSV): Optional visualization file specifying how to combine data and simulations for plotting. Different plots such as time-course or dose-response curves can be automatically created based on this file using the PETA Python library described below. This allows, for example, to quickly create visualizations to inspect parameter estimation results. A default visualization file can be automatically generated.

PETA problem file (YAML): File linking all of the above-mentioned PETA files together. This allows combinations of, e.g., multiple models or measurement files into a single parameter estimation problem, as well as easy reuse of various files in different parameter estimation problems (e.g., for model selection). The current YAML version 1.2 is used here.

We designed PETA to cover common features needed for parameter estimation. The TSV files comprise different mandatory columns. These provide all necessary information to define an objective function like the χ^2 or likelihood function. However, some methods tailored to specific problems require additional information to estimate the unknown parameters. To acknowledge this, we allow for optional application-specific extensions in addition to the required columns in the PETA files, e.g., if some parameters can be calculated analytically using hierarchical optimization approaches [18].

PETA library

To facilitate easy usability, PETA (<https://github.com/PETA-dev/PETA>) comes with detailed documentation describing the specific format of each of the different files in a concise yet comprehensive manner. Additionally, we provide a Python-based library that can be used to read, modify, write, and validate existing PETA problems. Furthermore, the PETA library provides functionality to package PETA files into COMBINE archives [19]. After parameter estimation, the modeler usually investigates how well the model fits the experimental data. To support this, the PETA library provides various visualization routines to analyze data and parameter estimation results.

Results

PEtab support in established tools

We implemented support for PEPtab in currently eight systems biology toolboxes, namely COPASI [2], AMICI [6], pyPESTO [20], pyABC [21], Data2Dynamics [5], dMod [10], parPE [18], and MEIGO [4]. These toolboxes provide a broad range of distinct features for model creation, model simulation, parameter inference, and uncertainty quantification (Table 1). Combining different tools with complementary features is often desirable. However, in practice this was hitherto hampered by the substantial overhead of tedious and error-prone re-implementation of the parameter estimation problem in the specific format required by the respective tool. With all of these tools now supporting PEPtab, a user can more easily combine different tools and make use of their specific strengths. For example, one can use COPASI for model creation and testing, AMICI for efficient simulation of large models, pyPESTO for multi-start local optimization and sampling, or MEIGO for global scatter searches, and Data2Dynamics or

Table 1. Non-exhaustive overview of the functionality offered by the different toolboxes currently supporting the PEPtab format. The list of supporting tools and functionality covered by the respective tools may increase over time. An updated overview is available on the PEPtab website. Darker colors indicate more accurate, scalable, or broader functionality compared to basic implementations.

	COPASI	D2D	dMod	MEIGO	AMICI	parPE	pyABC	pyPESTO
Interface / Language	Graphical interface	MATLAB	R	MATLAB	C++, Python, MATLAB	C++	Python	Python
Model construction	Advanced	Basic	Basic	No	No	No	No	No
Model simulation	Accurate	Accurate, Scalable	Accurate, Scalable	Uses AMICI	Accurate, Scalable	Uses AMICI	Uses AMICI	Uses AMICI
Gradient computation	Approximative	Accurate	Accurate	Uses AMICI	Accurate, Scalable	Uses AMICI	No	Uses AMICI
Gradient-free (global) parameter estimation	Multiple algorithms	Basic	No	Metaheuristic algorithms	No	No	No	Basic
Gradient-based parameter estimation	Basic	Multiple local optimizers	Multiple local optimizers	Metaheuristic algorithms	No	Multiple local optimizers	No	Multiple local optimizers
Parameter profile likelihood	Basic	Advanced	Advanced	No	No	No	No	Advanced
Prediction profile likelihood	Basic	Advanced	Advanced	No	No	No	No	No
Parameter sampling	No	Basic	No	No	No	No	Adaptive SMC algorithms	Multiple MCMC algorithms
Simulation of stochastic models	Multiple algorithms	No	No	No	No	No	No	No
Parameter inference for stochastic models	Basic	No	No	Basic	No	No	Scalable	Basic
Particular strengths	Advanced modeling Strong allrounder Graphical interface	Powerful gradient based optimization Advanced profiling Strong allrounder	Powerful gradient based optimization Strong allrounder	Powerful metaheuristic optimization	Highly scalable simulation & gradient comp.	Highly scalable optimization for large-scale models	Scalable likelihood-free inference	Allrounder Multiple MCMC sampling methods

<https://doi.org/10.1371/journal.pcbi.1008646.t001>

Table 2. Overview of supported PEPtab features in different tools, based on passed test cases of the PEPtab test suite. The first character indicates whether computing simulated data is supported and simulations are correct (✓) or not (-). The second character indicates whether computing χ^2 values of residuals are supported and correct (✓) or not (-). The third character indicates whether computing likelihoods is supported and correct (✓) or not (-). An up-to-date overview of supported features is maintained on the PEPtab GitHub page.

Test-case	AMICI	Copasi	D2D	dMod	MEIGO	parPE	pyABC	pyPESTO
Basic simulation	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Multiple simulation conditions	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Numeric initial compartment sizes in condition table	---	✓--	✓✓✓	✓✓✓	✓✓✓	---	---	---
Numeric initial concentration in condition table	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Numeric noise parameter overrides in measurement table	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Numeric observable parameter overrides in measurement table	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Observable transformations to log scale	✓-✓	✓--	✓✓✓	✓✓-	✓✓✓	--✓	✓-✓	✓-✓
Observable transformations to log10 scale	✓-✓	✓--	✓✓✓	✓✓-	✓✓✓	--✓	✓-✓	✓-✓
Parametric initial concentrations in condition table	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Parametric noise parameter overrides in measurement table	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Parametric observable parameteroverrides in measurement table	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Parametric overrides in condition table	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Partial pre-equilibration	✓✓✓	---	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Pre-equilibration	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Replicate measurements	✓✓✓	✓--	✓✓✓	✓✓✓	✓✓✓	--✓	✓✓✓	✓✓✓
Time-point specific overrides in the measurement table	---	---	✓✓✓	✓✓✓	✓✓✓	---	---	---

<https://doi.org/10.1371/journal.pcbi.1008646.t002>

dMod for profiling. The ease of switching between tools also provides the opportunity to easily reproduce and verify results, e.g., whether different tools yield similar results. Additionally, developers can compare the performance of newly developed methods with existing algorithms implemented in different toolboxes, independent of the programming language, to select the most appropriate one for a given setting.

PEtab test suite and examples

Along with introducing PEPtab support to different tools, we have set up a test suite with various toy problems and reference values that can be used by other tool developers to assess and verify PEPtab support in their software packages. The specific status of the PEPtab support of the different tools is provided in Table 2 and continuously updated on the PEPtab GitHub webpage. The test cases are based on SBML level 2 version 4 which is supported by all considered toolboxes.

To demonstrate the various features and the broad applicability of PEPtab, we provide a growing collection of currently 20 example parameter estimation problems in the PEPtab format largely based on a previously published benchmark collection [22]. These models can be used as templates for creating new PEPtab problems and for method development and testing.

Availability and future directions

PEtab complements existing standards for model definition by facilitating the specification of complex estimation problems using tabular text files, defining experimental measurements and linking model entities and measurements via observables and a noise model.

The specification of the PEPtab format, the PEPtab Python library, as well as links to examples, a web-based validation tool, and all supporting software are available at <https://github.com/PEtab-dev/PEtab>. A snapshot is available at <https://doi.org/10.5281/zenodo.3732958>. PEPtab and all original content presented here is available under permissive licences. For any

questions or requests related to PÉtab, we encourage interested users to approach us via the Issues function in the aforementioned GitHub repository, or the respective tool repositories for more specific queries.

We developed PÉtab to cover the most common features needed for parameter estimation in the context of dynamic modeling. However, as multiple model formats as well as a multitude of tailored parameter estimation methods exist, which require different information, we could not cover every aspect. While at the time of writing, PÉtab only allows for models defined in the SBML format, the PÉtab format is general enough to be integrated with other model specification formats like CellML and rule-based formats [13] in the future. Additionally, other formats like SÉtab [17] or Antimony [23] provide converters to SBML and can therefore also indirectly be used together with PÉtab. Recently, new methods have been developed to estimate parameters in a hierarchical manner [18], including from qualitative data [24, 25]. PÉtab could be extended to also allow for these types of measurements. To cover the most important needs, we invite users and developers to suggest new features to be supported by PÉtab. We formed a maintainer team comprising developers of all supporting toolboxes to facilitate long-term support and improvement of PÉtab. We encourage additional toolbox developers to implement support for PÉtab. As an example, since the preprint publication of this manuscript, PÉtab has already been adopted as the input format for a newly developed tool, SBML2Julia [26].

As PÉtab is already supported by software tools with hundreds of users in total, we envisage that it will facilitate reusability, reproducibility and interoperability. We expect that a common specification format will prove helpful for users as well as developers of parameter estimation tools and methods in systems biology.

Supporting information

S1 File. PÉtab specification. Detailed format description of PÉtab version 1. (PDF)

S2 File. PÉtab tutorial. Step-by-step instructions for creating PÉtab files for an application example. (PDF)

Acknowledgments

We thank Dagmar Waltemath for helpful discussions.

Author Contributions

Conceptualization: Leonard Schmiester, Yannik Schälte, Jan Hasenauer, Daniel Weindl.

Methodology: Leonard Schmiester, Yannik Schälte, Frank T. Bergmann, Fabian Fröhlich, Jan Hasenauer, Daniel Weindl.

Software: Leonard Schmiester, Yannik Schälte, Frank T. Bergmann, Tacio Camba, Erika Dudkin, Janine Egert, Fabian Fröhlich, Lara Fuhrmann, Adrian L. Hauber, Svenja Kemmer, Polina Lakrisenko, Carolin Loos, Simon Merkt, Wolfgang Müller, Dilan Pathirana, Elba Raimúndez, Lukas Refisch, Marcus Rosenblatt, Paul L. Stapor, Philipp Städter, Dantong Wang, Franz-Georg Wieland, Julio R. Banga, Jens Timmer, Alejandro F. Villaverde, Sven Sahle, Clemens Kreutz, Jan Hasenauer, Daniel Weindl.

Supervision: Wolfgang Müller, Julio R. Banga, Jens Timmer, Alejandro F. Villaverde, Clemens Kreutz, Jan Hasenauer.

Writing – original draft: Leonard Schmiester, Yannik Schälte, Jan Hasenauer, Daniel Weindl.

Writing – review & editing: Leonard Schmiester, Yannik Schälte, Frank T. Bergmann, Tacio Camba, Erika Dudkin, Janine Egert, Fabian Fröhlich, Lara Fuhrmann, Adrian L. Hauber, Svenja Kemmer, Polina Lakrisenko, Carolin Loos, Simon Merkt, Wolfgang Müller, Dilan Pathirana, Elba Raimúndez, Lukas Refisch, Marcus Rosenblatt, Paul L. Stapor, Philipp Städter, Dantong Wang, Franz-Georg Wieland, Julio R. Banga, Jens Timmer, Alejandro F. Villaverde, Sven Sahle, Clemens Kreutz, Jan Hasenauer, Daniel Weindl.

References

1. Kitano H. Computational Systems Biology. *Nature*. 2002; 420(6912):206–210. <https://doi.org/10.1038/nature01254>
2. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, et al. COPASI—a COmplex PATHway SIMulator. *Bioinformatics*. 2006; 22(24):3067–3074. <https://doi.org/10.1093/bioinformatics/btl485> PMID: 17032683
3. Balsa-Canto E, Banga JR. AMIGO, a toolbox for advanced model identification in systems biology using global optimization. *Bioinformatics*. 2011; 27(16):2311–2313. <https://doi.org/10.1093/bioinformatics/btr370>
4. Egea JA, Henriques D, Cokelaer T, Villaverde AF, MacNamara A, Danciu DP, et al. MEIGO: An open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinf*. 2014; 15(136). <https://doi.org/10.1186/1471-2105-15-136> PMID: 24885957
5. Raue A, Steiert B, Schelker M, Kreutz C, Maiwald T, Hass H, et al. Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems. *Bioinformatics*. 2015; 31(21):3558–3560. <https://doi.org/10.1093/bioinformatics/btv405> PMID: 26142188
6. Fröhlich F, Kaltenbacher B, Theis FJ, Hasenauer J. Scalable Parameter Estimation for Genome-Scale Biochemical Reaction Networks. *PLoS Comput Biol*. 2017; 13(1):e1005331. <https://doi.org/10.1371/journal.pcbi.1005331>
7. Choi K, Medley JK, König M, Stocking K, Smith L, Gu S, et al. Tellurium: An extensible python-based modeling environment for systems and synthetic biology. *Bio Systems*. 2018; 171:74–79. <https://doi.org/10.1016/j.biosystems.2018.07.006> PMID: 30053414
8. Stapor P, Weindl D, Ballnus B, Hug S, Loos C, Fiedler A, et al. PESTO: Parameter ESTimation TOolbox. *Bioinformatics*. 2018; 34(4):705–707. <https://doi.org/10.1093/bioinformatics/btx676> PMID: 29069312
9. Mitra ED, Suderman R, Colvin J, Ionkov A, Hu A, Sauro HM, et al. PyBioNetFit and the Biological Property Specification Language. *iScience*. 2019; 19:1012–1036. <https://doi.org/10.1016/j.isci.2019.08.045> PMID: 31522114
10. Kaschek D, Mader W, Fehling-Kaschek M, Rosenblatt M, Timmer J. Dynamic Modeling, Parameter Estimation, and Uncertainty Analysis in R. *J Stat Softw*. 2019; 88(10). <https://doi.org/10.18637/jss.v088.i10>
11. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, et al. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*. 2003; 19(4):524–531. <https://doi.org/10.1093/bioinformatics/btg015> PMID: 12611808
12. Cuellar AA, Lloyd CM, Nielsen PF, Bullivant DP, Nickerson DP, Hunter PJ. An Overview of CellML 1.1, a Biological Model Description Language. *Simulation*. 2003; 79(12):740–747. <https://doi.org/10.1177/0037549703040939>
13. Harris LA, Hogg JS, Tapia JJ, Sekar JAP, Gupta S, Korsunsky I, et al. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics*. 2016; 32(21):3366–3368. <https://doi.org/10.1093/bioinformatics/btw469> PMID: 27402907
14. Waltemath D, Adams R, Bergmann FT, Hucka M, Miller FKAK, Moraru II, et al. Reproducible computational biology experiments with SED-ML—The Simulation Experiment Description Markup Language. *BMC Syst Biol*. 2011; 5(198). <https://doi.org/10.1186/1752-0509-5-198> PMID: 22172142
15. Choi K, Smith LP, Medley JK, Sauro HM. phraSED-ML: A paraphrased, human-readable adaptation of SED-ML. *Journal of bioinformatics and computational biology*. 2016; 14(06):1650035. <https://doi.org/10.1142/S0219720016500359>
16. Dada JO, Spasić I, Paton NW, Mendes P. SBRML: a markup language for associating systems biology data with models. *Bioinformatics*. 2010; 26:932–938. <https://doi.org/10.1093/bioinformatics/btq069>

17. Lubitz T, Hahn J, Bergmann FT, Noor E, Klipp E, Liebermeister W. SBtab: a flexible table format for data exchange in systems biology. *Bioinformatics*. 2016; 32(16):2559–2561. <https://doi.org/10.1093/bioinformatics/btw179>
18. Schmiester L, Schälte Y, Fröhlich F, Hasenauer J, Weindl D. Efficient parameterization of large-scale dynamic models based on relative measurements. *Bioinformatics*. 2019; 36(2):594–602.
19. Bergmann FT, Adams R, Moodie S, Cooper J, Glont M, Golebiewski M, et al. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC bioinformatics*. 2014; 15:369. <https://doi.org/10.1186/s12859-014-0369-z> PMID: 25494900
20. Schälte Y, Fröhlich F, Stapor P, Wang D, Weindl D, Schmiester L, et al. ICB-DCM/pyPESTO: pyPESTO 0.0.11; 2020. Available from: <https://doi.org/10.5281/zenodo.3715448>.
21. Klinger E, Rickert D, Hasenauer J. pyABC: distributed, likelihood-free inference. *Bioinformatics*. 2018; 34(20):3591–3593. <https://doi.org/10.1093/bioinformatics/bty361>
22. Hass H, Loos C, Raimúndez-Álvarez E, Timmer J, Hasenauer J, Kreutz C. Benchmark problems for dynamic modeling of intracellular processes. *Bioinformatics*. 2019; 35(17):3073–3082. <https://doi.org/10.1093/bioinformatics/btz020>
23. Smith LP, Bergmann FT, Chandran D, Sauro HM. Antimony: a modular model definition language. *Bioinformatics*. 2009; 25(18):2452–2454. <https://doi.org/10.1093/bioinformatics/btp401>
24. Mitra ED, Dias R, Posner RG, Hlavacek WS. Using both qualitative and quantitative data in parameter identification for systems biology models. *Nature communications*. 2018; 9(1):3901. <https://doi.org/10.1038/s41467-018-06439-z>
25. Schmiester L, Weindl D, Hasenauer J. Parameterization of mechanistic models from qualitative data using an efficient optimal scaling approach. *J Math Biol*. 2020; 81(2):603–623. <https://doi.org/10.1007/s00285-020-01522-w> PMID: 32696085
26. Lang PF, Shin S, Zavala VM. SBML2Julia: interfacing SBML with efficient nonlinear Julia modelling and solution tools for parameter optimization. *arXiv preprint arXiv:2011.02597*. 2020.