The Journal of Open Source Software

LikelihoodProfiler.jl: Unified profile-likelihood workflows for identifiability and confidence intervals

Ivan Borisov [1], **Aleksander Demin**[2], **and Evgeny Metelkin** [1]

1 InSysBio CY LTD, Limassol, Cyprus **2** National Research University Higher School of Economics, Moscow, Russia

DOI: 10.21105/joss.09501

Software

- Review
- Repository
- Archive

Editor: Owen Lockwood

Reviewers:

- @salbalkus
- @currocam
- @dufourc1

Submitted: 04 September 2025
Published: 20 January 2026

Summary

Mathematical models — mechanistic, statistical, or otherwise — contain unknown parameters that must be estimated from data to enable reliable predictions. Profile likelihood is a widely used approach for studying the *practical identifiability* of such models, that is, assessing whether the available data constrain parameter estimates tightly enough to yield finite confidence intervals. Applicable to a broad class of *maximum-likelihood estimation* problems, profile likelihood is especially valuable for complex nonlinear models — such as high-dimensional mechanistic models — where standard asymptotic confidence intervals can be unreliable. Profile likelihood techniques can also be generalized to analyze derived quantities such as model states and predictions. This versatility makes profile likelihood an essential component in model development and uncertainty assessment (Heinrich et al., 2025).

`LikelihoodProfiler.jl` is an open-source Julia package that implements several profile likelihood methods through a unified interface, enabling parameters and prediction uncertainty analysis.

Statement of Need

Researchers across many scientific domains need tools to quantify how well estimates of model parameters and predictions are constrained by data. Profile likelihood methods provide an intuitive and interpretable approach to parameter and prediction uncertainty, but existing software implementations - such as `sbioparameterci (MATLAB)` (The MathWorks, Inc., 2024), `dMod (R)` (Kaschek et al., 2019), and `pyPESTO (Python)` (Schälte et al., 2025) - are often tied to specific modeling ecosystems or limited to a single profiling method. Within the Julia ecosystem, packages such as `ProfileLikelihood.jl` (VandenHeuvel, 2021) and `InformationGeometry.jl` (Arutjunjan & Beyer, 2025) provide related functionality but focus on specific profile likelihood methods rather than offering multiple profiling strategies through a unified interface. As a result, users may need to combine multiple tools or reimplement profiling routines.

`LikelihoodProfiler.jl` addresses this need by providing a unified, model-agnostic interface for computing profile likelihoods for parameters and arbitrary functions of parameters. The package offers several profiling methods - such as optimization-based `OptimizationProfiler`, integration-based `IntegrationProfiler`, confidence-interval endpoint search via `CICOProfiler` - and integrates with the Julia SciML (Rackauckas & Nie, 2017), making profile likelihood analysis accessible and efficient for a wide variety of applied problems.

Demonstrative Example: JAK/STAT Signaling Pathway Model

`LikelihoodProfiler.jl`'s functionality is demonstrated using the example from Systems Biology: JAK/STAT signaling pathway ODE model (Boehm et al., 2014), which consists of 8 states and 9 parameters. The model and experimental data were sourced from the

Benchmark-Models-PEtab repository (Contributors, 2024) and imported through the `PEtab.jl` package (Persson et al., 2025).

```
using LikelihoodProfiler, Optimization, OptimizationLBFGSB,
      OrdinaryDiffEqTsit5, PEtab, CICOBase, Plots

petab_model = PEtabModel("Boehm_JProteomeRes2014.yaml")
petab_problem = PEtabODEProblem(petab_model)
```

To study the identifiability of the JAK/STAT model parameters, we first construct a `ProfileLikelihoodProblem`. `ProfileLikelihoodProblem` is defined by providing (i) the objective function (typically the negative log-likelihood) and (ii) parameter values corresponding to the optimum of this objective. `LikelihoodProfiler.jl` builds on the `Optimization.jl` interface (Dixit & Rackauckas, 2023), and `ProfileLikelihoodProblem` wraps an `OptimizationProblem`. In addition, `ProfileLikelihoodProblem` allows users to specify optional arguments, such as the indices of parameters to profile, upper and lower bounds, and other options.

```
optprob = OptimizationProblem(petab_problem)
optpars = Vector(get_x(petab_problem))
param_profile_prob = ProfileLikelihoodProblem(optprob, optpars; idxs=1:3)
```

`LikelihoodProfiler.jl` offers a suite of methods to solve the `ProfileLikelihoodProblem`. In this example, we use the Hessian-free variant of the `IntegrationProfiler` (Chen & Jennrich, 2002), which approximates the likelihood profile and performs re-optimization after each ODE solver step to prevent divergence from the true profile trajectory. Each profiling method offers several configurable options.

All methods use the common `CommonSolve.solve()` interface (Rackauckas & Nie, 2017), supporting global settings for parallelization, verbosity, and initialization strategies.

```
alg_integ = IntegrationProfiler(integrator=Tsit5(),
  integrator_opts = (dtmax=0.5, reltol=1e-3, abstol=1e-4),
  matrix_type = :identity, gamma=0., reoptimize=true,
  optimizer = LBFGSB(),optimizer_opts=(maxiters=10000,))

sol_param = solve(param_profile_prob, alg_integ, verbose=true)
```

An alternative approach, implemented in `CICOProfiler`, estimates the confidence intervals (CI) endpoints directly — without reconstructing the full profile — by solving a constrained optimization problem (Borisov & Metelkin, 2020). This method is often more efficient when only the CI is required.

```
alg_cico = CICOProfiler(optimizer = :LN_NELDERMEAD, scan_tol = 1e-10)
```

Users can get estimated confidence intervals via `endpoints(sol_param)` and identifiability status via `retcodes(sol_param)`. Profile curves can be visualized using `Plots.jl` (`plot(sol_param)`) or exported as DataFrames.

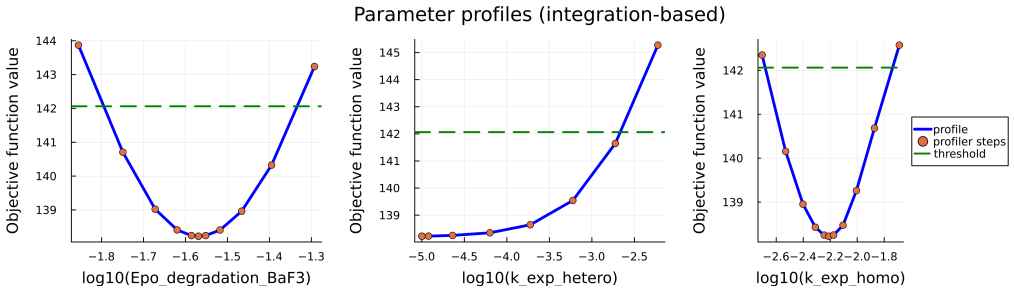

Figure 1: Parameter profiles for the JAK/STAT model computed using an integration-based profiler

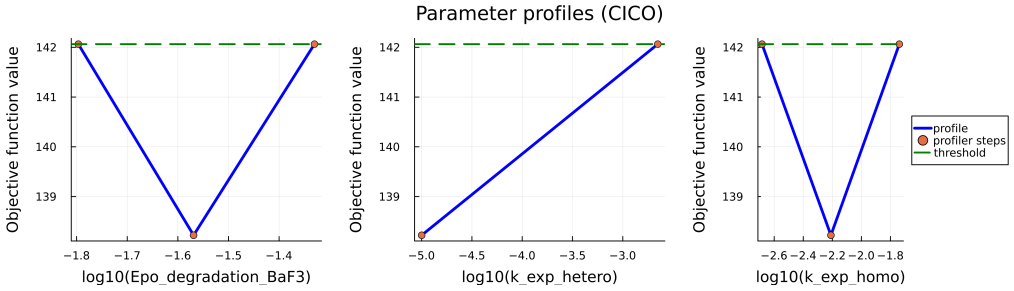

Figure 2: Parameter profiles for the JAK/STAT model computed using the CICO method

The same algorithms can also be applied to arbitrary functions of parameters. This generalization of profile likelihood concept (Kreutz et al., 2012) can be used to study model reparametrization or perform predictability analysis. This functionality is available through the same interface: users define functions of parameters and provide them as the third argument to the problem. For illustration, we define a function that returns the observable value at selected time points:

```
function pSTAT5A_rel_obs(x, p, t)
  sol = get_odesol(x, petab_problem)(t)
  specC17 = 0.107
  pSTAT5A_rel = (100 * sol[7] + 200 * sol[6] * specC17) /
    (sol[7] + sol[1] * specC17 + 2 * sol[6] * specC17)
  return pSTAT5A_rel
end
pSTAT5A_rel_optf(t) = OptimizationFunction(
  (x,p) -> pSTAT5A_rel_obs(x,p,t), AutoFiniteDiff())

times = [2.5, 60.0, 200.0]
func_profile_prob = ProfileLikelihoodProblem(optprob, optpars,
 [pSTAT5A_rel_optf(t) for t in times];
 profile_lower=0.0, profile_upper=120.0)

sol_func = solve(func_profile_prob, alg_integ)
```

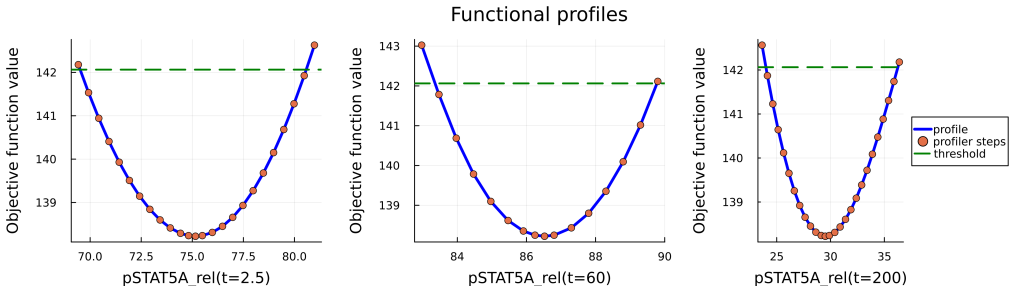

Figure 3: Functional profiles of a JAK/STAT model observable evaluated at three time points

Prediction confidence interval endpoints can be estimated at a larger set of time points, and smooth prediction bands can then be plotted.

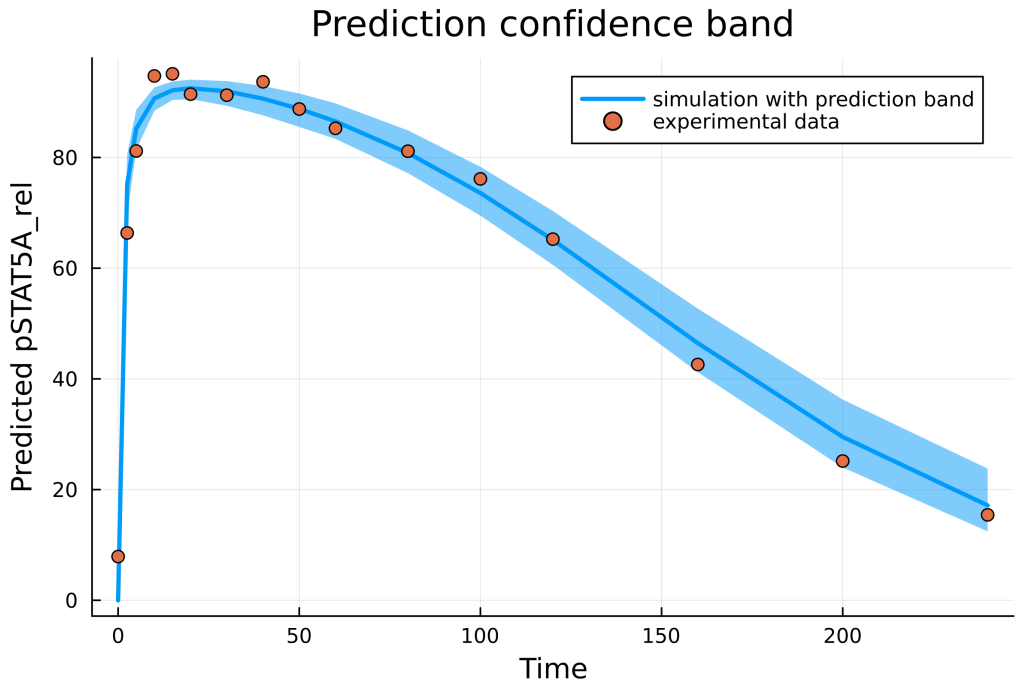

Figure 4: Prediction confidence band for a JAK/STAT model observable across multiple time points

Implementation and Extensibility

All profiling methods benefit from the unified interface provided by `LikelihoodProfiler.jl`:

- Integration with `SciML` packages provides access to multiple optimizers, differential equation solvers, and AD backends.
- Compatibility with `Heta` (Metelkin, 2021) and `PEtab` formats broadens the accessibility of the package across different modeling frameworks.
- A single `solve()` interface enables use of profiling methods for both parameters and functions.
- A common parallelization setup, controlled via the `parallel_type` argument in the `solve()` can significantly accelerate computations.
- The interface facilitates integration of new profiling methods and stepping strategies.

Future work will include adding new methods of parameters, functions and predictions profiling and enabling adaptive switching between strategies.

Acknowledgements

This study was carried out as part of the authors' work at InSysBio CY LTD. No specific external funding was received.

References

Arutjunjan, R., & Beyer, S. (2025). *RafaelArutjunjan/InformationGeometry.jl: v1.22.3* (Version v1.22.3). Zenodo. https://doi.org/10.5281/zenodo.17882740

Boehm, M. E., Adlung, L., Schilling, M., Roth, S., Klingmüller, U., & Lehmann, W. D. (2014). Identification of isoform-specific dynamics in phosphorylation-dependent STAT5 dimerization by quantitative mass spectrometry and mathematical modeling. *J. Proteome Res.*, *13*(12), 5685–5694. https://doi.org/10.1021/pr5006923

Borisov, I., & Metelkin, E. (2020). Confidence intervals by constrained optimization—an algorithm and software package for practical identifiability analysis in systems biology. *PLoS Comput Biol*, *16*(12), e1008495. https://doi.org/10.1371/journal.pcbi.1008495

Chen, J.-S., & Jennrich, R. I. (2002). Simple accurate approximation of likelihood profiles. *Journal of Computational and Graphical Statistics*, *11*(3), 714–732. https://doi.org/10.1198/106186002493

Contributors, T. Pe. B. C. (2024). *The PEtab Benchmark Collection of parameter estimation problems* (Version 2024.10.15). Zenodo. https://doi.org/10.5281/zenodo.8155057

Dixit, V. K., & Rackauckas, C. (2023). *Optimization.jl: A unified optimization package* (Version v3.12.1). Zenodo. https://doi.org/10.5281/zenodo.7738525

Heinrich, M., Rosenblatt, M., Wieland, F.-G., Stigter, H., & Timmer, J. (2025). On structural and practical identifiability: Current status and update of results. *Current Opinion in Systems Biology*, *50*, 100546. https://doi.org/10.1016/j.coisb.2025.100546

Kaschek, D., Mader, W., Fehling-Kaschek, M., Rosenblatt, M., & Timmer, J. (2019). Dynamic modeling, parameter estimation, and uncertainty analysis in R. *Journal of Statistical Software*, *88*(10), 1–32. https://doi.org/10.18637/jss.v088.i10

Kreutz, C., Raue, A., & Timmer, J. (2012). Likelihood based observability analysis and confidence intervals for predictions of dynamic models. *BMC Systems Biology*, *6*(120), 1–12. https://doi.org/10.1186/1752-0509-6-120

Metelkin, E. (2021). Heta compiler: A software tool for the development of large-scale QSP models and compilation into simulation formats. *JOSS*, *6*(67), 3708. https://doi.org/10.21105/joss.03708

Persson, S., Fröhlich, F., Grein, S., Loman, T., Ognissanti, D., Hasselgren, V., Hasenauer, J., & Cvijovic, M. (2025). PEtab.jl: Advancing the efficiency and utility of dynamic modelling. *Bioinformatics*, *41*(9), btaf497. https://doi.org/10.1093/bioinformatics/btaf497

Rackauckas, C., & Nie, Q. (2017). DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in Julia. *JORS*, *5*(1), 15. https://doi.org/10.5334/jors.151

Schälte, Y., Fröhlich, F., Stapor, P., Vanhoefer, J., Weindl, D., Jost, P. J., Wang, D., Lakrisenko, P., Raimúndez, E., Pathirana, D., Schmiester, L., Städter, P., Contento, L., Merkt, S., Dudkin, E., Grein, S., & Hasenauer, J. (2025). *pyPESTO - parameter EStimation TOolbox for Python* (Version v0.5.7). Zenodo. https://doi.org/10.5281/zenodo.17603051

The MathWorks, Inc. (2024). *SimBiology toolbox: sbioparameterci*. https://www.mathworks.com/help/simbio/ref/sbioparameterci.html

VandenHeuvel, D. (2021). *ProfileLikelihood.jl*. https://github.com/DanielVandH/

ProfileLikelihood.jl