LikelihoodProfiler.jl: A Unified Package for Practical Identifiability Analysis and Confidence Intervals Estimation

Ivan Borisov[1], Aleksander Demin[1], Evgeny Metelkin[1]

[1] InSysBio

GitHub

INTRODUCTION

Practical identifiability addresses the critical question of how well a mechanistic model is determined by the available experimental data. In many cases, profile likelihood-based methods are used as a proxy for structural identifiability analysis, particularly when the complexity of a model makes structural methods inapplicable or computationally prohibitive.

Moreover, profile likelihood techniques can be extended beyond parameter analysis to assess the identifiability of model states and predictions. This versatility makes profile likelihood analysis an essential component in the development and validation of models in **Systems Biology** and **Quantitative Systems Pharmacology (QSP)**.

OBJECTIVE

This study presents **LikelihoodProfiler.jl,** an **open-source Julia package** for conducting **profile likelihood-based identifiability and uncertainty analysis**. The package provides a **unified, extensible interface** to multiple profiling methods and integrates seamlessly with Julia's scientific computing ecosystem.

This study demonstrates the application of **several profiling methods** on a model of the **JAK/STAT signaling pathway** [2], which consists of 8 state variables and 9 parameters.

METHODS

The model structure, experimental data, and simulation conditions were obtained from the **Benchmark-Models-PEtab** GitHub repository and imported into Julia using the **PEtab.jl package** [3].

```
using Petab, Plots

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

To define the profile likelihood problem, we construct an `OptimizationProblem` instance and provide the optimal parameter values:

```
using Optimization, LikelihoodProfiler

optprob = OptimizationProblem(petab_problem)
plprob = PLProblem(optprob, get_x(petab_problem))
```

LikelihoodProfiler.jl offers **a suite of methods for profiling likelihood functions and assessing practical identifiability**. Each method includes several configurable options, such as optimizer or integrator selection, tolerances, and step size.

The most straightforward method is `OptimizationProfiler`, which follows the classical approach of **stepwise re-optimization of the likelihood function** under a constraint on the parameter of interest.

```
method1 = OptimizationProfiler(optimizer = Optimization.LBFGS(),
stepper = FixedStep(; initial_step=0.07))
```

A more advanced method is the `IntegrationProfiler`, which computes likelihood profiles by **solving a system of differential equations** derived from the underlying optimization problem. This method requires a differential equation solver (`integrator`) to be specified.

```
using OrdinaryDiffEq

method2 = IntegrationProfiler(integrator = Tsit5(),
integrator_opts = (dtmax=0.07,), matrix_type = :hessian)
```

An alternative approach, implemented in `CICOProfiler`, **estimates the confidence intervals (CI) endpoints directly**—without reconstructing the full profile—by solving a constrained optimization problem [1]. This method is often more efficient when only the CI is required.

```
using CICOBase

method3 = CICOProfiler(optimizer = :LN_NELDERMEAD,
scan_tol = 1e-10)
```

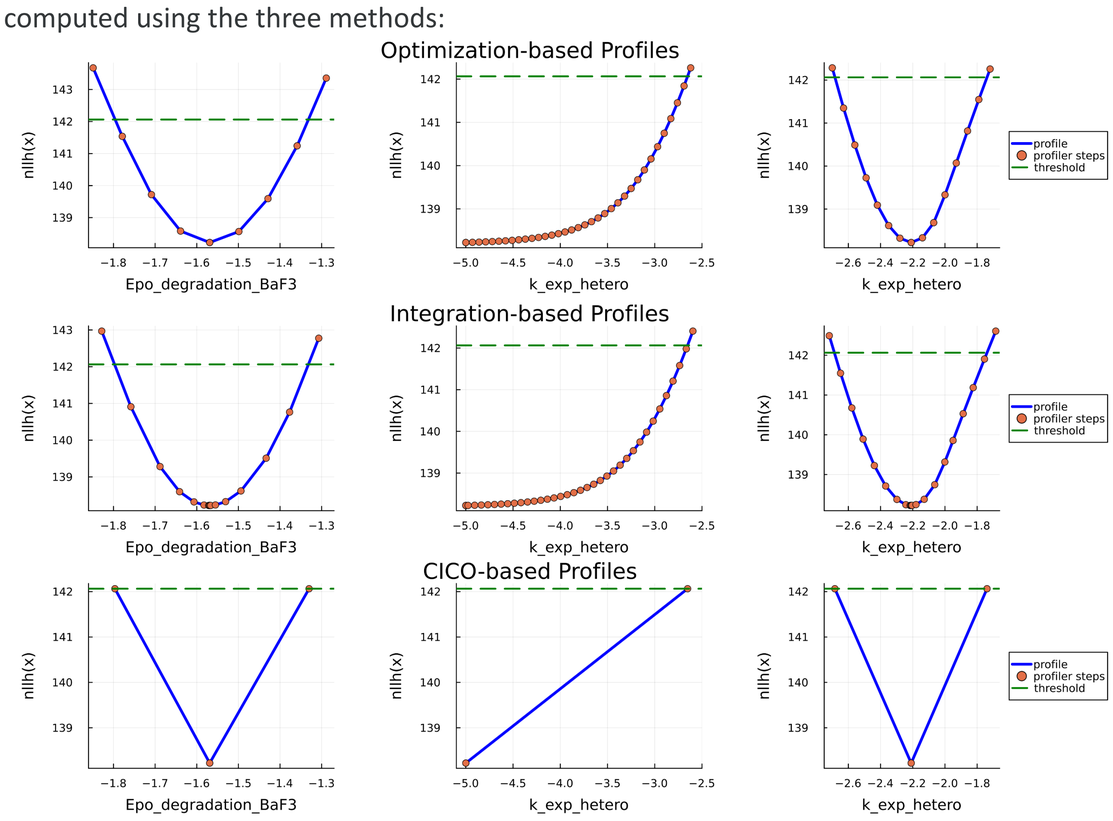
All profiling methods share a common `profile()` interface, allowing users to specify global settings (`kwargs`) such as **verbosity and parallelization options**.

```
sol = profile(plprob, method; kwargs…)
```

Profiling results can be **visualized using the Plots.jl** package or **exported as a DataFrame** for further analysis.

RESULTS

Below are the profile likelihoods for the first three parameters of the JAK/STAT model, computed using the three methods:



All three methods **reported similar CI** for the JAK/STAT model, which can be accessed using the `get_endpoint()` function.

Parameter	OptimizationProfiler	IntegrationProfiler	CICOProfiler
Epo_degradation_BaF3	(-1.796, -1.332)	(-1.797, -1.332)	(-1.797, -1.33)
k_exp_hetero	(nothing, -2.653)	(nothing, -2.652)	(nothing, -2.652)
k_exp_homo	(-2.683, -1.739)	(-2.687, -1.739)	(-2.684, -1.739)
k_imp_hetero	(-1.947, -1.626)	(-1.947, -1.622)	(-1.949, -1.622)
k_imp_homo	(-0.109, nothing)	(-0.109, nothing)	(-0.11, nothing)
k_phos	(4.084, 4.331)	(4.08, 4.331)	(4.078, 4.332)
sd_pSTAT5A_rel	(0.365, 0.871)	(0.363, 0.869)	(0.362, 0.871)
sd_pSTAT5B_rel	(0.626, 1.084)	(0.627, 1.084)	(0.623, 1.085)
sd_rSTAT5A_rel	(0.312, 0.763)	(0.311, 0.763)	(0.308, 0.764)

The optimal profiling method and settings depend on the complexity of the model and the goal of the analysis:

- `OptimizationProfiler` benefits from the **choice of optimization algorithm** (e.g., gradient-based or derivative-free) but may be **computationally intensive**.
- `IntegrationProfiler` provides **smooth profile trajectories** but requires Hessian computation or approximation, which may be **challenging for large-scale models**.
- `CICOProfiler` is often more **efficient for CI estimation** when the full profile is not needed.

CONCLUSIONS

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

- **Integration with SciML packages** gives users access to a wide range of optimizers, differential equation solvers, and AD backends, enabling efficient profiling configurations.
- **Compatibility with Heta, PEtab and SBML formats** broadens the accessibility of the package across different modeling frameworks.
- **A common parallelization setup**, controlled via the `parallel_type` argument in the `profile()` function, is **supported across all methods** and can significantly accelerate computations.
- The interface facilitates **integration of new profiling methods and stepping algorithms**.

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

REFERENCES

[1] I. Borisov et al., "Confidence intervals by constrained optimization—An algorithm and software package for practical identifiability analysis in systems biology". PLoS Comput Biol. 2020.

[2] M.E. Boehm et al. "Identification of Isoform-Specific Dynamics in Phosphorylation-Dependent STAT5 Dimerization by Quantitative Mass Spectrometry and Mathematical Modeling." Journal of Proteome Research. 2014.

[3] S. Persson et al., "PEtab.jl: Advancing the Efficiency and Utility of Dynamic Modelling," (preprint)

CONTACTS

For more information, please visit our web site at **www.insysbio.com**.
Ivan Borisov by e-mail: **borisov@insysbio.com**