The Journal of Open Source Software

folie: Finding Optimal Langevin Inferred Equations

Hadrien Vroylandt [1,2¶], **Daniele Bersano**[3], **and Jérôme Hénin**[3¶]

1 Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ, GREYC UMR 6072, 14000 Caen, France **2** Sorbonne Université, Institut des sciences du calcul et des données, ISCD - F-75005 Paris, France **3** Université Paris Cité, CNRS, Laboratoire de Biochimie Théorique UPR 9080, 75005, Paris, France ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 23 September 2025
Published: unpublished

Summary

This paper introduces FOLIE (Finding Optimal Langevin Inferred Equations), a versatile Python library designed to facilitate the analysis of high-dimensional molecular simulation trajectories in terms of low-dimensional dynamics. FOLIE enables scientists to fit low-dimensional stochastic differential equations (SDEs) to projected high-dimensional data, thereby extracting maximum dynamical insight from the simulations. Key features of FOLIE include robust estimation techniques, comprehensive analysis tools, and simulation capabilities to create synthetic datasets. Its highly modular architecture allows for the implementation of diverse SDEs, time discretization methods, and energy landscapes. By leveraging FOLIE, researchers can effectively extrapolate low-dimensional kinetics from limited simulation data, enhancing their ability to understand and predict complex molecular dynamics.

Statement of need

Quantitative predictions of rare events in molecular and materials simulations suffer from the curse of dimensionality and the prohibitive cost of simulating relevant time scales, starting from microscopic time steps (on the order of $10^{-15}s$). Such rare events include chemical reactions in chemistry, drug unbinding in pharmacology, and phase transitions in materials. A strategy to make such predictions tractable is to construct intermediate, low-dimensional kinetic models of the time evolution of the system. Such low-dimensional models are constructed based on a projection of the full dynamics onto a reduced set of collective variables. This set must fulfill two constraints: describe the process of interest, and be informative enough to capture the long-time dynamics of this process.

FOLIE is designed to allow easy and efficient inference of such models from projected molecular simulations. There exits several software packages performing related tasks, but FOLIE differ mainly by its flexibility in the description of the energy landscape and its modular construction for the estimation task. DeepTime (previously pyEmma) (Hoffmann et al., 2021) is an equivalent for discrete Markovian processes. pymle fits continuous SDEs but imposes restraints on the underlying energy landscapes, which are better suited to econometrics and financial markets (Kirkby et al., 2025). OptLE (Palacio-Rodriguez & Pietrucci, 2022) is an experimental Fortran package that inspired this work. It focuses on method development and was not designed for flexibility or scalability. pyOptLE was a first attempt at improving scalability, with a very limited scope. StochasticForceInference and UnderdampedLangevinInference are codes from the same group focusing on the similar task of fitting continuous SDE but focusing on biological applications and such having less flexibility on the underlying energy landscapes.

The Journal of Open Source Software

Theoretical background

Langevin Models

Let's consider the dynamics of a low-dimensional collective variable q. There is a set of possible Langevin models to describe this dynamicsGirardier et al. (2023). Projecting the high-dimensionnal dynamics onto the collective variable leads to the generalized Langevin equation(Vroylandt, 2022), that writes for a single variable

$$\ddot{q} = -\frac{1}{m(q)}\frac{\partial A(q)}{\partial q} + k_B T \frac{\partial m(q)^{-1}}{\partial q} - \int_0^t \Gamma(s)\dot{q}(t-s)\,\mathrm{d}s + R(t)$$

Here, $m(q)$ represent a position-dependent effective mass, $-\frac{\partial A(q)}{\partial q}$ is the conservative force field in which the dynamics takes place, and the effective free energy surface is $A(q) = -k_B T \log(\rho_{eq}(q))$ where ρ_{eq} is the invariant distribution of the dynamics. $\Gamma(s)$ is a memory kernel, a time dependent function describing the correlation of the velocity at time t with itself at a previous time $t-s$ and $R(t)$ is a random force. The random force is usually assumed to be related to the memory kernel according to the fluctuation-dissipation theorem $\langle R(0)R(t)\rangle = \frac{k_B T}{m}\Gamma(t)$, with $m = \int m(q)\rho_{eq}(q)\mathrm{d}q$, even if this relation is approximate in this framework(Vroylandt, 2022).

Assuming that the timescale of evolution of the collective variable is slow with respect to its environment; we can take the assumption of a Dirac kernel $\Gamma(s) = \gamma\delta(s)$, the fluctuation dissipation theorem being now valid. We then obtain the memory-less (Markovian) Standard Langevin equation

$$\ddot{q} = -\frac{1}{m(q)}\frac{\partial A(q)}{\partial q} + k_B T \frac{\partial m(q)^{-1}}{\partial q} - \frac{\gamma}{m(q)}\dot{q} + \sqrt{\frac{2k_B T\gamma}{m(q)}}\eta(t) \tag{1}$$

where $\eta(t)$ is now a standard Gaussian noise. One common assumption that we do not take here is a position-independent effective mass. If one were to consider the dynamics for underdamped motion on a time scale $\tau \gg \frac{m}{\gamma}$ non equilibrium fluctuations are quickly damped. This entitles us to improperly consider $\ddot{q} \approx 0$ leading to the Overdamped Langevin Equation

$$\dot{q} = -\beta D(q)\frac{\partial A(q)}{\partial q} + \frac{\partial D(q)}{\partial q} + \sqrt{2D(q)}\eta(t) \tag{2}$$

with the definition of the diffusion profile from $D(q) = \frac{k_B T}{m(q)\gamma}$ and using $\beta = \frac{1}{k_B T}$. This last model sensibly simplify the mathematical structure being a first order differential equation. The current state of the library is to infer reduced Langevin models starting from projected simulation trajectories, focusing so far on the overdamped case.

Kinetic model optimization by a maximum-likelihood approach

In order to construct the optimal Langevin model to describe the dynamics in this lower-dimensional projection, we start by defining $\mathcal{L}(\vec{q}|\theta)$, the likelihood of observing the trajectory data $\vec{q}$ if they were to be generated by one of the specified Langevin models parameterized by θ. Here θ stands for a compact way of regrouping both the drift $F(q) = -\beta D(q)\frac{\partial A(q)}{\partial q} + \frac{\partial D(q)}{\partial q}$ and position dependent diffusion $D(q)$. Consequently the best values of θ is found by maximizing the likelihood of the observed trajectory

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ \mathcal{L}(\theta). \tag{3}$$

The analytical shape of the likelihood is not always known a priory but, restricting to the overdamped case, it will be the product of short time transition probability between consecutive

trajectory points due to the Markovian property of the equation itself (Palacio-Rodriguez & Pietrucci, 2022). Using the transition density $p_\theta(q_{i+1}, t_{i+1}|q_i, t_i)$, i.e. the probability of being in position q_i at time t_i starting from q_{i-1} at time t_{i-1} , for given initial conditions (q_0, t_0), the likelihood can be written as

$$\mathcal{L}(\vec{q}|\theta) = \prod_{i=0}^{N-1} p_\theta(q_{i+1}, t_{i+1}|q_i, t_i) \tag{4}$$

from which follows that the log-likelihood of the trajectory is :

$$\log \mathcal{L}(\vec{q}|\theta) = \sum_{i=0}^{N-1} \log \left[p_\theta(q_{i+1}, t_{i+1}|q_i, t_i)\right] \tag{5}$$

The precise form of the transition density is contingent on the choice of a specific time discretization (essentially related to the choice of an integrator) of the continuous SDE, for which several possibilities are implemented within FOLIE as explained below.

Implementation

Features

The library has been implemented with a modular structure, so that users can assemble the relevant components in simple Python scripts tailored to their needs.

- **Model of Overdamped Langevin Dynamics** Several models of possible implementation of Overdamped Langevin equations are present starting from the parent python class `Overdamped`, followed by the implementation of particular cases such as `BrownianMotion` and `OrnsteinUhlenbeck`. The underdamped case is still under development.
- **Model for the force and diffusion coefficient functions** In addition to the dynamics, building the transition density and likelihood estimator requires the specification of the drift and space-dependent diffusion coefficient. To that effect, the `Function` class has been implemented, which offers functional forms such as polynomials, splines, etc.
- **Transition densities** Different methods aimed at approximating the form of the propagator are implemented(Iacus, 2008). They require as input a `Model` object whose drift and diffusion will be used to compute the mean, variance, and, in case of the Elerian transition probability density, the additional parameters to obtain the relative likelihood. These probability densities are later fed to the `Likelihood` estimator object, which optimizes the drift and diffusion parameters.
- **Estimation** Given as input the probability densities used to compute the likelihood of the observed trajectories, an estimator object is created. The class of estimator objects playing a central role in performing the addressed task is the `LikelihoodEstimator` class, within it, the MLE estimators are recovered by making use of the optimize.minimize() method from the scipy library applied to the negative of the log-likelihood function eq.(5).
- **Simulation** In addition to the principal purpose of training the maximum-likelihood estimator for a given set of input trajectories, the FOLIE module also allows to simulate trajectories, which is useful for creating synthetic data to be used when developing methods. This is achieved by first specifying the model of the Langevin equation guiding the evolution of the system through a suitable `Overdamped` object. Then this passed to the `Simulator` (or possibly `BiasedSimulator`) class specifying the integration timestep employed. Finally, the dataset is generated by calling `Simulator.run()`.

Initial guess for parameters

Optimization of the likelihood requires an initial guess for the drift and diffusion parameter. In FOLIE, we use a Kramers-Moyal estimation to provide such a guess. The Kramers-Moyal estimator consists of the empirical estimation of the first two coefficients of the Kramers-Moyal expansion for the evolution (Master) equation associated with the equilibrium probability density $\rho_{eq}(q)$, *i.e* the Fokker-Plank equation (Risken, 1996).

From a a set of trajectories, the Kramers-Moyal estimator computes the drift term (respectively the diffusion term) as the first (respectively second) moment of the displacement conditioned on the position. We obtain the parameters θ from the two equations

$$F^{KM}(q) = \langle (q_{i+1} - q_i)|q \rangle / \Delta t, \tag{6}$$

$$D^{KM}(q) = \langle (q_{i+1} - q_i - F^{KM}(q_i)\Delta t)^2|q \rangle / \Delta t. \tag{7}$$

It is worth noticing that the Kramers-Moyal procedure leads to the correct parameters in the limit of a small timestep Δt and a sufficient number of trajectories passing at position q. In which case, it becomes equivalent to a maximum-likelihood estimation using Euler discretization of the propagator. It thus provides a good starting guess for the maximum-likelihood estimator.

Parallel computation

One key motivation behind the writing of folie was performance. Likelihood calculation scales linearly with the data size, and with the model complexity, making optimization potentially slow. Furthermore, Langevin optimization can be integrated in a scheme for optimizing collective variables (Mouaffac et al., 2023), in which case a large number of model optimizations must be performed before collective variable optimization converges. When running folie in a shared-memory multiprocessor environment, likelihood computation is performed in a data-parallel way over the projected simulation trajectories.

Practical use

Usage workflow

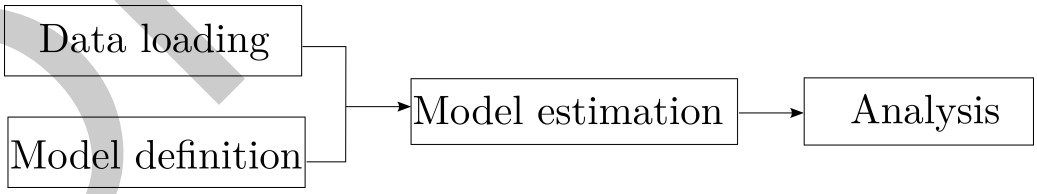

Figure 1: Typical workflow in folie. This basic workflow is illustrated in this example script in the repository. Several options are available for the model definition and the model estimation.

Perspectives

Further developments are in progress, in particular more widely applicable forms of dynamics, namely underdamped and generalized Langevin dynamics. Thanks to the modular design of FOLIE, these will integrate seamlessly into the workflow.

Acknowledgements

We are indebted to Fabio Pietrucci for spearheading the scientific effort that led us to develop FOLIE. We acknowledge stimulating discussions with Arthur France-Lanord, David Girardier,

Léo Hallegot, Léon Huet, and Line Mouaffac.

References

Girardier, D. D., Vroylandt, H., Bonella, S., & Pietrucci, F. (2023). Inferring free-energy barriers and kinetic rates from molecular dynamics via underdamped Langevin models. *The Journal of Chemical Physics*, *159*(16), 164111. https://doi.org/10.1063/5.0169050

Hoffmann, M., Scherer, M. K., Hempel, T., Mardt, A., Silva, B. de, Husic, B. E., Klus, S., Wu, H., Kutz, J. N., Brunton, S., & Noé, F. (2021). Deeptime: A python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*.

Iacus, S. M. (2008). *Simulation and Inference for Stochastic Differential Equations: With R Examples* (Vol. 1). Springer. https://doi.org/10.1007/978-0-387-75839-8

Kirkby, J., Nguyen, D., Nguyen, D., & Nguyen, N. N. (2025). Pymle: A python package for maximum likelihood estimation and simulation of stochastic differential equations. *Journal of Statistical Software, Forthcoming*.

Mouaffac, L., Palacio-Rodriguez, K., & Pietrucci, F. (2023). Optimal Reaction Coordinates and Kinetic Rates from the Projected Dynamics of Transition Paths. *Journal of Chemical Theory and Computation*, *19*(17), 5701–5711. https://doi.org/10.1021/acs.jctc.3c00158

Palacio-Rodriguez, K., & Pietrucci, F. (2022). Free energy landscapes, diffusion coefficients, and kinetic rates from transition paths. *Journal of Chemical Theory and Computation*, *18*(8), 4639–4648. https://doi.org/10.1021/acs.jctc.2c00324

Risken, H. (1996). *Fokker-planck equation*. Springer Berlin Heidelberg. https://doi.org/https://doi.org/10.1007/978-3-642-61544-3

Vroylandt, H. (2022). On the derivation of the generalized Langevin equation and the fluctuation-dissipation theorem. *Europhysics Letters*, *140*(6), 62003. https://doi.org/10.1209/0295-5075/acab7d