The Journal of Open Source Software

dcTMD: a python package for performing dissipation-corrected targeted molecular dynamics

Miriam Jäger[1], **Victor Tänzel**[1], **Daniel Nagel**[2], **and Steffen Wolf**[1]

1 University of Freiburg, Institute of Physics, Freiburg, Germany **2** University of Heidelberg, Institute of Theoretical Physics, Heidelberg, Germany

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

dcTMD is a Python package designed to extract free-energy and nonequilibrium friction estimates from targeted molecular dynamics (TMD) simulations (Schlitter et al., 1994). The method implemented here is called *dissipation-corrected targeted molecular dynamics* (dcTMD) by Wolf & Stock (2018). Given a set of non-equilibrium simulations where, for example, a ligand is pulled from a binding site via a velocity constraint, this tool performs automated post-processing of the bias-force time traces to estimate the underlying free-energy landscape and the friction (dissipation) along the unbinding coordinate.

The method is based on a second-order cumulant expansion of *Jarzynski's equality* (Jarzynski, 1997), which connects nonequilibrium work distributions to equilibrium free-energy differences. Combined with a Markovian Langevin Equation, dcTMD further allows the extraction of *position- and velocity-dependent friction coefficients* from the same nonequilibrium data. This approach has been succesfully applied in multiple studies (Wolf et al. (2020) Jäger et al. (2022) Post et al. (2022) Tänzel et al. (2024) Jäger & Wolf (2025) Milster et al. (2025)). The resulting free-energy and friction profiles can subsequently be used to estimate *binding and unbinding rate constants* following Wolf et al. (2020).

The software is intended for molecular dynamics practitioners interested in ligand–protein unbinding, mechanistic interpretation of binding kinetics, and quantitative modeling of non-equilibrium effects in soft condensed matter and biomolecular systems.

Statement of need

Ligand unbinding from proteins is of fundamental interest in computational biophysics (Schuetz et al. (2017)). In many cases, the unbinding event is rare and requires enhanced-sampling or biased-simulation strategies to observe within computationally feasible timescales. dcTMD-based workflows have been shown to yield accurate free energy and non-equilibrium friction coefficients from velocity-constrained pulling simulations. The dcTMD package builds on this work by offering a unified, documented, and extensible implementation that is currently not available, thereby lowering the barrier for applying dcTMD to new biomolecular systems and for reproducing published dcTMD studies. By providing a dedicated Python framework with an `scikit-learn`-style API, `dcTMD` enables users to integrate dissipation-corrected analysis into existing workflows, ensuring reproducibility and broad accessibility. The software has already been successfully applied in several studies (e.g. (Tänzel et al., 2024), (Jäger & Wolf, 2025)), and is expected to promote the wider adoption of the dissipation-corrected targeted MD approach in computational chemistry and biophysics.

The Journal of Open Source Software

Implementation and architecture

The code is written in Python (versions 3.9–3.14) and is available under the MIT license.
Repository: https://github.com/moldyn/dcTMD

Key architectural features include:

- A modular API following `fit/transform` conventions familiar from *scikit-learn*, easing integration into analysis pipelines.
- Input support for *GROMACS* pulling trajectories.
- Core functionality for computing free energy and non-equilibrium friction profiles along the biasing coordinate.
- Support for analysis of multiple unbinding pathways.
- Force correlation analysis for non-equilibrium friction analysis.
- Continuous integration and testing via GitHub Actions; documentation hosted at https://moldyn.github.io/dcTMD.

Use case

A typical workflow begins with the user performing at least 100 independent velocity-constraint pulling simulations. `dcTMD` provides two analysis routes, both following the same workflow pattern:

1. Work-based analysis using a `WorkSet` and `WorkEstimator`
2. Force-correlation analysis using a `ForceSet` and `ForceEstimator`

Both methods yield free-energy and friction profiles but differ in how these properties are estimated.

1. Load trajectories into a `WorkSet` or `ForceSet`

The user loads all pulling trajectories into an appropriate container:

- `WorkSet` for the work-based route, which is computationally cheaper, as the resolution of the trajectories can be reduced after integration.
- `ForceSet` for the force-correlation route.

Each trajectory contains the constraint force $f(t)$ from which the work along the pulling coordinate is computed as $W(x) = \int_{x_0}^{x} \mathrm{d}x' f(x')$.

2. Perform dcTMD analysis via an estimator

- **Work-based estimator (`WorkEstimator`)** The free-energy profile is estimated as $\Delta G(x) = \langle W(x) \rangle - \frac{\beta}{2}\langle \delta W(x)^2 \rangle$, with $\delta W = W - \langle W \rangle$, $\beta = (k_\mathrm{B}T)^{-1}$, and $\langle . \rangle$ denoting a trajectory ensemble mean. The dissipated work is $W_\mathrm{diss}(x) = \frac{\beta}{2}\langle \delta W(x)^2 \rangle$. The non-equilibrium position-dependent friction is obtained from its derivative as $\Gamma(x) = \frac{1}{v}\frac{\mathrm{d}}{\mathrm{d}x} W_\mathrm{diss}(x)$.
- **Force-correlation-based estimator (`ForceEstimator`)** In this approach, ΔG and Γ are computed directy from the force data as $\Delta G(x) = \int_{x_0}^{x} \mathrm{d}x' \langle f(x') \rangle - v \int_{x_0}^{x} \mathrm{d}x' \Gamma(x')$ and $\Gamma(x) = \beta \int_0^{t(x)} \mathrm{d}\tau \langle \delta f(t(x)) \delta f(\tau) \rangle$. The two-time force autocorrelation function $C_t(\tau) = \langle \delta f(t(x)) \delta f(\tau) \rangle$ can be plotted to gain insight into timescales within degrees of freedom orthogonal to x.

3. Visualize

`dcTMD` provides plotting tools for work distribution analysis, free-energy $\Delta G(x)$ and friction profiles $\Gamma(x)$.

The Journal of Open Source Software

4. Example

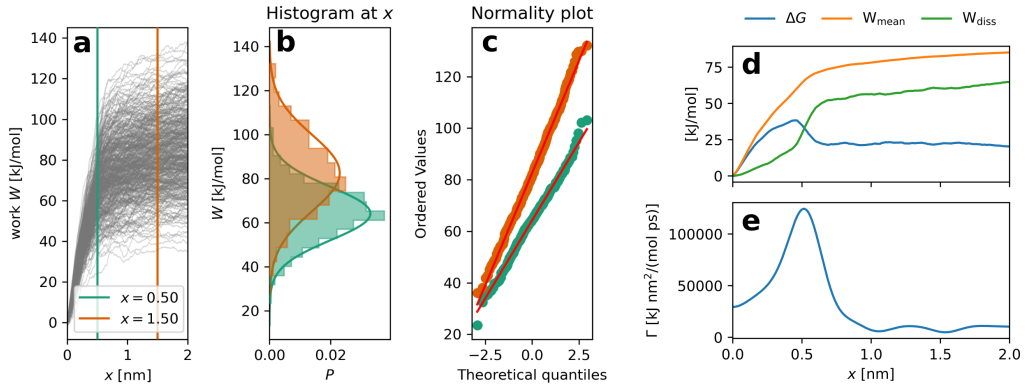

Figure 1: Figures crated using data taken from Wolf et al. (2020) of trypsin-benzamidine unbinding. a)-c) work distribution analysis. d) Decomposition of mean work $W_{\mathrm{mean}} = \langle W(x) \rangle$ into free energy $\Delta G(x)$ and dissipation work $W_{\mathrm{diss}}(x)$. e) non-equilibrium friction coefficient $\Gamma(x)$ along the pulling coordinate x.

Figure 1 displays a common analysis of a set of unbinding trajectories from TMD simulations of the trypsin-benzamidine complex (Wolf et al., 2020). The analysis of the work distribution displays good agreement with a normal distribution at two different evaluated positions of the pulling coordinate x. The mean work $W_{\mathrm{mean}} = \langle W(x) \rangle$, which shows no features on its own, yields a free energy profile $\Delta G(x)$, which displays a clearly defined transition state at $x \approx 0.45$ nm as well as a bound state in form of a free energy minimum at $x \approx 0.0$ nm and an unbound continuum for $x > 0.6$ nm. The maximum in friction Γ around $x = 0.5$ nm is indicative of changes in the hydration of both ligand and binding site.

Impact

By providing an open and reproducible implementation of the dcTMD methodology, the software lowers the barrier for researchers to apply dissipation-corrected targeted molecular dynamics to ligand unbinding problems as well as other condensed soft matter systems. This enables broader exploration of nonequilibrium binding kinetics, supports mechanistic interpretation of frictional contributions, and provides access to position-dependent friction from targeted MD trajectories. We anticipate the tool will be adopted in academic molecular simulation groups and in pharmaceutical research exploring unbinding free energies and kinetics.

Acknowledgements

The implementation of dcTMD builds on the Python scientific stack, relying on **NumPy** (Harris et al., 2020) for numerical operations, **Matplotlib** (Hunter, 2007) for visualization, and **Click** (The Pallets Projects, 2023) for the command-line interface. We thank Gerhard Stock, Matthias Post and Georg Diez for valuable discussions, and Fabian Rohrbach and Leo Küchler for testing the software. This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) via the Research Unit FOR 5099 *"Reducing complexity of nonequilibrium systems"* (project No. 431945604). The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen and the Rechenzentrum of the University of Freiburg, as well as the state of Baden-Württemberg through bwHPC and the DFG through Grants Nos. INST 37/935-1 FUGG and INST 39/963-1 FUGG.

References

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*, 90–95. https://doi.org/10.1109/MCSE.2007.55

Jäger, M., Koslowski, T., & Wolf, S. (2022). Predicting ion channel conductance via dissipation-corrected targeted molecular dynamics and langevin equation simulations. *J. Chem. Theory Comput.*, *18*(1), 494–502. https://doi.org/10.1021/acs.jctc.1c00426

Jäger, M., & Wolf, S. (2025). More sophisticated is not always better: A comparison of similarity measures for unsupervised learning of pathways in biomolecular simulations. *J. Phys. Chem. B*, *129*(42), 10956–10966. https://doi.org/10.1021/acs.jpcb.5c04586

Jarzynski, C. (1997). Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, *78*(14), 2690–2693. https://doi.org/10.1103/PhysRevLett.78.2690

Milster, S., Dzubiella, J., Stock, G., & Wolf, S. (2025). Nonequilibrium friction and free energy estimates for kinetic coarse-graining—driven particles in responsive media. *J. Chem. Phys.*, *162*(15), 154113. https://doi.org/10.1063/5.0261459

Post, M., Wolf, S., & Stock, G. (2022). Molecular origin of driving-dependent friction in fluids. *J. Chem. Theory Comput.*, *18*(5), 2816–2825. https://doi.org/10.1021/acs.jctc.2c00190

Schlitter, J., Engels, M., & Krüger, P. (1994). Targeted molecular dynamics: A new approach for searching pathways of conformational transitions. *J. Mol. Graph.*, *12*(2), 84–89. https://doi.org/10.1016/0263-7855(94)80072-3

Schuetz, D. A., Witte, W. E. A. de, Wong, Y. C., Knasmueller, B., Richter, L., Kokh, D. B., Sadiq, S. K., Bosma, R., Nederpelt, I., Heitman, L. H., Segala, E., Amaral, M., Guo, D., Andres, D., Georgi, V., Stoddart, L. A., Hill, S., Cooke, R. M., Graaf, C. de, … Ecker, G. F. (2017). Kinetics for drug discovery: An industry-driven effort to target drug residence time. *Drug Discov. Today*, *22*(6), 896–911. https://doi.org/10.1016/j.drudis.2017.02.002

Tänzel, V., Jäger, M., & Wolf, S. (2024). Learning protein–ligand unbinding pathways via single-parameter community detection. *J. Chem. Theory Comput.*, *20*(12), 5058–5067. https://doi.org/10.1021/acs.jctc.4c00250

The Pallets Projects. (2023). *Click: Python composable command line interface toolkit* (Version 8.1.*). https://palletsprojects.com/p/click

Wolf, S., Lickert, B., Bray, S., & Stock, G. (2020). Multisecond ligand dissociation dynamics from atomistic simulations. *Nat. Commun.*, *11*(1), 2918. https://doi.org/10.1038/s41467-020-16655-1

Wolf, S., & Stock, G. (2018). Targeted molecular dynamics calculations of free energy profiles using a nonequilibrium friction correction. *J. Chem. Theory Comput.*, *14*(12), 6175–6182. https://doi.org/10.1021/acs.jctc.8b00835