



Classical MD E-CAM Modules I

E-CAM Deliverable 1.2

Deliverable Type: Report

Delivered in December, 2017



E-CAM

The European Centre of Excellence for
Software, Training and Consultancy
in Simulation and Modelling



Funded by the European Union under grant agreement 676531

Project and Deliverable Information

| | |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Project Title | E-CAM: An e-infrastructure for software, training and discussion in simulation and modelling |
| Project Ref. | Grant Agreement 676531 |
| Project Website | https://www.e-cam2020.eu |
| EC Project Officer | Juan Pelegrín |
| Deliverable ID | D1.2 |
| Deliverable Nature | Report |
| Dissemination Level | Public |
| Contractual Date of Delivery | Project Month 16(1 st February, 2017) |
| Actual Date of Delivery | 7 th December, 2017 |
| Description of Deliverable | 9 software modules in classical Molecular Dynamics delivered to the E-CAM repository in the area of trajectory sampling, thermodynamics and kinetics of rare events, and their documentation. |

Document Control Information

| | | |
|------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Document | Title: | Classical MD E-CAM Modules I |
| | ID: | D1.2 |
| | Version: | As of 7 th December, 2017 |
| | Status: | Accepted by WP leader |
| | Available at: | https://www.e-cam2020.eu/deliverables |
| Review | Document history: | Internal Project Management Link |
| | Review Status: | Reviewed |
| Authorship | Action Requested: | Submit |
| | Written by: | David W.H. Swenson (University of Amsterdam) |
| | Contributors: | Christoph Dellago (University of Vienna), Peter G. Bolhuis (University of Amsterdam), Jony Castagna (STFC, Sci-Tech Daresbury Laboratory), Sonya M. Hanson (Memorial Sloan Kettering Cancer Center) |
| | Reviewed by: | Alan Ó Cais (DE-JUELICH) and Ana Mendonça (EPFL) |
| | Approved by: | Christoph Dellago (Universität Wien) |

Document Keywords

| | |
|-----------|-------------------------------------------------------------------------------------------------------------------|
| Keywords: | E-CAM, Molecular Dynamics, Module, Rare Events, Path Sampling, Open-PathSampling, Kinetics, Thermodynamics, CECAM |
|-----------|-------------------------------------------------------------------------------------------------------------------|

7th December, 2017

Disclaimer: This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements.

Copyright notices: This deliverable was co-ordinated by David W.H. Swenson¹ (University of Amsterdam) on behalf of the E-CAM consortium with contributions from Christoph Dellago (University of Vienna), Peter G. Bolhuis (University of Amsterdam), Jony Castagna (STFC, Sci-Tech Daresbury Laboratory), Sonya M. Hanson (Memorial Sloan Kettering Cancer Center). This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0>.



¹dwhs@hyperblazer.net

Contents

| | |
|---------------------------------------------------------------------------------------------|-----------|
| Executive Summary | 1 |
| 1 Introduction | 2 |
| 1.1 Modules in E-CAM Work Package 1 | 2 |
| 1.2 This Deliverable in the context of E-CAM | 2 |
| 2 Background | 4 |
| 2.1 Rare events and path sampling | 4 |
| 2.2 Applications and exploitation of path sampling and the modules in this report | 4 |
| 2.3 The role of these modules in the path sampling workflow | 5 |
| 3 Modules | 7 |
| 3.1 Path density | 7 |
| 3.1.1 Module description | 7 |
| 3.1.2 Motivation and exploitation | 7 |
| 3.2 Direct (on-the-fly) flux and rate calculation | 7 |
| 3.2.1 Module description | 7 |
| 3.2.2 Motivation and exploitation | 8 |
| 3.3 Improved input for OPS networks | 8 |
| 3.3.1 Module description | 8 |
| 3.3.2 Motivation and exploitation | 9 |
| 3.4 New WHAM code for OPS | 9 |
| 3.4.1 Module description | 9 |
| 3.4.2 Motivation and exploitation | 9 |
| 3.5 Flux/Rate from existing trajectories | 10 |
| 3.5.1 Module description | 10 |
| 3.5.2 Motivation and exploitation | 10 |
| 3.6 OPS snapshot features | 10 |
| 3.6.1 Module description | 10 |
| 3.6.2 Motivation and exploitation | 11 |
| 3.7 Two-way shooting | 11 |
| 3.7.1 Module description | 11 |
| 3.7.2 Motivation and exploitation | 11 |
| 3.8 Annotated trajectories | 12 |
| 3.8.1 Module description | 12 |
| 3.8.2 Motivation and exploitation | 12 |
| 3.9 OPS Piggybacker | 12 |
| 3.9.1 Module description | 12 |
| 3.9.2 Motivation and exploitation | 12 |
| 4 Performance Considerations | 14 |
| 5 Outlook | 15 |
| References | 16 |

Executive Summary

In this report for Deliverable 1.2 of E-CAM, 9 software modules in classical dynamics are presented. These modules represent improvements and new features in software for path (trajectory) sampling and for studying the thermodynamics and kinetics of rare events. They include aspects relating to all stages of performing a simulation of a rare event, including the simulation setup, running the simulation, and analyzing the results of the simulation.

The selection of the modules reported here was motivated by several goals, including:

- To implement modules that were specifically recommended by [E-CAM Deliverable D1.1](#) (Identification/selection of E-CAM MD codes for development) [1] and by the [report of the E-CAM Classical MD State of the Art Workshop²](#), held 29 August – 2 September 2016 at the Lorentz Center in Leiden, Netherlands.
- To deliver modules that are prerequisites for the development of future modules, especially those to be developed at E-CAM Extended Software Development Workshops.
- To respond to the scientific needs of several active research projects, including projects related to conformational changes in DNA and in cancer-causing proteins. These projects are real-world use cases that illustrate how these modules can be exploited by academia and industry.

The modules in this report are based on the Python code [OpenPathSampling \(OPS\)](#). For cases where performance is critical, these modules, like the rest of OPS, inherit the simulation performance and scaling from the underlying molecular dynamics (MD) engine that the code wraps around.

Future E-CAM classical MD modules will build on the modules reported here, in order to enable more complex algorithms, including the six modules developed at the Extended Software Development Workshop (ESDW) held in Traunkirchen, Austria, in November 2016.

Of the 9 modules presented here, 7 have been incorporated into the core code of OPS. They are:

- path density
- direct (on-the-fly) flux and rate calculation
- improved input for OPS networks
- new WHAM code for OPS
- flux/rate from existing trajectories
- OPS snapshot features
- two-way shooting

The other 2 modules build on OpenPathSampling, but remain separate. They are:

- `annotated_trajectories`
- `ops_piggybacker`

Each module is thoroughly tested with automated software tests, and includes in-code documentation as well as external documentation in the form of [Jupyter notebook](#) examples.

Section 1 of this report gives a brief description of E-CAM modules (in general) and the role of this deliverable in the broader goals of E-CAM Work Package 1 (WP1). Section 2 provides some brief background material on rare events, the applications of path sampling methods to study rare events, and the role of these modules within the practice of path sampling. In section 3, we describe each of the modules: In addition to a description of the module, for each module there is a subsection explaining the motivation and exploitation of the module, and there are links to further material on the E-CAM module documentation websites. That material provides detailed information about the code development, testing and documentation of the modules. Section 4 describes performance aspects of these modules, and section 5 summarizes and describes the outlook for future development of modules within WP1.

²Report available for download from the E-CAM website: <https://www.e-cam2020.eu/scientific-reports/>

1 Introduction

Work Package 1 (WP1) of E-CAM provides a means for academic and industrial users to address questions that involve classical MD by developing open source software with verified quality standards, appropriate documentation, and testing.

Computational classical MD is a mature field, with many packages for molecular simulation. However, there is still a lack of widely distributed software for more advanced techniques based on MD, such as sampling trajectories for rare events. E-CAM WP1 will develop software to fill this gap.

1.1 Modules in E-CAM Work Package 1

The software E-CAM develops is presented as “modules,” a term that is used here in a broader sense than is typically used in the semantics of high-level programming languages. E-CAM modules can include workflow scripts, analysis tools and test suites, as well as traditional subroutines and functions. Importantly, these modules adhere to best practices in programming style conventions, in-code documentation, and regression/unit testing. They are written so that they can take advantage of anticipated hardware developments in the near future (one of the training objectives of E-CAM).

Information about all the modules is collected on a website, which is intended to be a simple way for the public to browse these modules. For WP1, the documentation website is <http://e-cam-classical-md-modules.readthedocs.io>. This documentation itself is developed in an open source manner, with transparent development and a willingness to accept contributions from the public, in a repository on the E-CAM GitLab service. For WP1, this documentation is developed at <https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules>. Much of the text in this report is drawn from materials generated for that website.

Since the modules themselves often contribute to external software packages, the module documentation serves as a central repository for metadata about the modules. In addition to a description of the modules, it includes links to the specific code developed as part of each module, and to examples, either in the external package’s documentation or separate from it. It also gives potential users information about how to install the software and run tests to ensure that it works.

1.2 This Deliverable in the context of E-CAM

This report covers the first nine modules delivered as a part of E-CAM WP1. As described in the grant agreement, these first modules are “in the area of trajectory sampling, thermodynamics and kinetics of rare events.” The modules described herein were primarily developed by the E-CAM WP1 Postdoctoral Research Associate (PDRA), between March 2016 and January 2017.

Trajectory sampling (also called “path sampling”) is a family of methods for studying the thermodynamics and kinetics rare events. In [E-CAM Deliverable D1.1](#) [1], we provided an overview of existing software for rare events, and found that OpenPathSampling (OPS) was the optimal choice for E-CAM development. OPS is an open-source Python package for path sampling that wraps around other classical MD codes. In section 4.2 of D1.1, we highlighted several areas where E-CAM could make useful contributions to OPS, including the addition of two-way shooting and path density plots, which are included as part of this Deliverable. The annotated trajectories module is an example of a state definition assessor, which was identified as a target for E-CAM development in section 4 of the report³ from the State of the Art Workshop held in Leiden in 2016.

Code for the seven modules that were incorporated into the core of OPS is stored as pull requests on the official [OpenPathSampling GitHub repository](#). The other two, the [annotated trajectories module](#) and the [OPSPiggybacker module](#), are hosted on their developer’s GitHub page. Additional examples for several modules are hosted on E-CAM’s GitLab server, in the [ops_additional_examples repository](#). Both GitHub and GitLab use powerful version control systems for software development, although E-CAM’s own GitLab Server allows more control of software development, for example, the private hosting of the code.

In the next section, we will describe the problem of rare events, and how path sampling methods are used to study the thermodynamics and kinetics of rare events, as well providing an overview of how the modules we developed contribute to the overall process of running a path sampling simulation. In section 3, we describe each of the individual modules delivered as part of this report, along with links to the E-CAM documentation, which in turn contains links to

³Report available for download from the E-CAM website: <https://www.e-cam2020.eu/scientific-reports/>

examples, further documentation, and the source code for the module. Section 4 will describe the role of high performance computing in these modules, and section 5 concludes and provides outlook for future modules to be delivered by E-CAM WP1.

2 Background

2.1 Rare events and path sampling

In many simulations, we come across the challenge of bridging timescales. The desire for high resolution in space (and therefore time) is inherently in conflict with the desire to study long-time dynamics. To study molecular dynamics with atomistic detail, we must use timesteps on the order of a femtosecond. However, many problems in biological chemistry, materials science, and other fields involve events that only spontaneously occur after a millisecond or longer (for example, biomolecular conformational changes, or nucleation processes). That means that we would need around 10^{12} time steps to see a single millisecond-scale event. This is the problem of “rare events” in theoretical and computational chemistry and physics.

While modern supercomputers are beginning to make it possible to obtain trajectories long enough to observe some of these processes (such as [millisecond dynamics of a protein](#) [2]), even then, we may only find one example of a given transition. To fully characterize a transition (with proper statistics), we need many examples. This is where path sampling comes in. Path sampling approaches obtain many trajectories using a Markov chain Monte Carlo approach: An existing trajectory is perturbed (usually using a variant of the “shooting” move), and the resulting trial trajectory is accepted or rejected according to conditions that preserve the distribution of the path ensemble. As such, path sampling is Monte Carlo in the space of paths (trajectories). Conceptually, this enhances the sampling of transitions by focusing on the transition region instead of the stable states. In direct MD, trajectories spend much more time in stable states than in the transition region (exponential population differences for linear free energy differences); path sampling skips over that time in the stable states.

The main path sampling approaches used in the modules below are [transition path sampling \(TPS\)](#) [3] and [transition interface sampling \(TIS\)](#) [4]. In practice, TPS is mainly used to characterize the mechanism of a transition, while TIS (which is more expensive than TPS) is used to calculate rates and free energy landscapes. Overviews of these methods, as well as other rare events methods, can be found in the following review articles:

- [2010 review by Bolhuis and Dellago in Reviews in Computational Chemistry](#) [5]
- [2008 review by Dellago and Bolhuis in Advances in Polymer Science](#) [6]

In addition, several other resources are available on the web to teach path sampling, including:

- [Wikipedia entry on path sampling](#) [7]
- [Aaron Keys’s tutorial on path sampling](#) [8]

2.2 Applications and exploitation of path sampling and the modules in this report

Since the problem of bridging timescales, which path sampling addresses, is a generic one, path sampling can be used in many fields. Indeed, there’s nothing in the methodology that even restricts it to molecular simulation. However, it is best known in the field of classical MD simulations, where path sampling methods have shown many successes, including:

- [Mechanisms of complex chemical reactions, such as autoionization of water](#) [9]
- [Mechanism of hydrophobic assembly](#) [10]
- [Evidence that the glass transition is a first-order phase transition](#) [11]
- [Mechanism of crystal nucleation](#) [12]
- [Mechanism of cavitation](#) [13]
- [Identifying new mechanisms in catalytic systems](#) [14]
- [Characterization of the conformational dynamics networks in proteins](#) [15]

As computational resources become more powerful, path sampling has the promise to provide insight into rare events in larger systems, and into events with even longer timescales. For example:

- Drug/protein binding and unbinding (timescales of minutes), which is essential for predicting the efficacy of drugs
- Association processes of proteins (large systems), which is at the core of communication in biochemical pathways

- Self assembly processes for complex systems (many intermediates), which can be important for the design of new materials

Further, applying the known successes of path sampling methods to larger systems can also be quite valuable. Path sampling can shed light on the networks of conformational dynamics for large proteins and protein complexes, and on the mechanisms and rates of complex reactions and phase transitions. The range of possibilities is so broad that it is impossible to enumerate – both academics and industry will benefit greatly from having software for these methods.

The modules in this report were selected for development in response to the needs of several scientific projects, as well as plans for the E-CAM WP1 ESDW held in Traunkirchen in November 2016. The roles that specific modules played in these scientific projects will be described with each module in section 3. However, as a brief overview, the scientific projects that motivated the software modules documented in this report included the following:

- **WC-HG DNA.** In addition to the well-known Watson-Crick (WC) base pairing motif, recent research has shown that an alternative conformation, known as the Hoogsteen (HG) motif, is also significant at physiological conditions [16]. It has been shown that the HG motif plays an important role in some DNA replication processes, and it may be relevant to many other biological processes as well. Path sampling methods (TPS and TIS) and modules developed in E-CAM are used to study the rate and mechanism of the transitions between the WC and HG base pairing motifs in DNA. This work was carried out by researchers at the University of Amsterdam, and is currently in preparation for publication [17].
- **KRas dynamics.** Mutant forms of the KRas protein are implicated in many cancers, including 71% of pancreatic cancers [18]. Experimental studies suggest that, when KRas is GTP bound, it can switch between a less flexible, more active state, and a more flexible, less active state. The active state is associated with gain of function mutations, which result in accelerated cell growth and proliferation [19].

Path sampling methods and modules developed by E-CAM made it possible to identify differences between the dynamics of wild-type GTP-bound KRas and an oncogenic mutant. In particular, TPS was used to study the transition between the less flexible and more flexible states. A better understanding of this mechanism could direct research into new and effective drugs targetting proteins in the Ras family, which have been notoriously difficult drug targets. The simulations were performed with OPS, and used several of the modules included in this report. This work included a master's thesis project and a bachelor's thesis project, both performed at the University of Amsterdam. It is currently in preparation for publication [20].

- **DFG flip in kinases.** Kinases are one of the most targeted families of proteins by the pharmaceutical industry. The drug imatinib was designed to inhibit Abl kinase, and became the first targeted cancer therapy. Imatinib binds a specific conformation known as the DFG-out conformation of kinases, and now many kinase inhibitors have been specifically designed to fit in a pocket that opens when the phenylalanine of the DFG flips to the “out” conformation. Understanding the kinetics and thermodynamics of this process has significant implications toward the selective design of kinase-targeted therapies [21]. With over 500 human kinases, many of which are implicated in disease, and all of which have the conserved DFG motif, computational methods will be key. A module included in this report was used to analyze data generated by the Folding@Home project on Src and Abl kinases, as part of a collaboration between the University of Amsterdam and Memorial Sloan Kettering Cancer Center.
- **OPS introduction.** The modules in this report are based on the OpenPathSampling software package, which is still in development for its version 1.0 release. Several of the modules play fundamental roles in OPS, and will be important for the paper introducing the code, which is currently in preparation for publication [22].

2.3 The role of these modules in the path sampling workflow

The process of path sampling, like many types of simulation, can be split into three stages:

1. Set up the simulation
2. Run the simulation
3. Analyze the results of the simulation

The modules included in this report contribute to all three stages. In addition, we include modules that affect the underlying data structures of OPS, which play a more holistic role of connecting the three stages of the simulation.

Setting up a path sampling simulation is a more intricate process than setting up many other kinds of simulations. In fact, it often takes longer to set up a project than to run the simulation itself. This is because path sampling requires the definition of all relevant stable states of the system, and, for some types of simulations, an approximate

reaction coordinate and interface locations along that reaction coordinate. These parts of setting up a path sampling simulation are particularly difficult, and there was a lack of specialized software tools to help.

Several of the modules included in this report deal with ways to characterize stable states, and to determine whether interfaces are placed at appropriate locations. In particular, the annotated trajectories module is used to check state definitions. In addition, the modules that calculate the flux (both “on-the-fly” and from existing trajectories) can be used to check that the innermost interface is placed at an appropriate location. These are essential steps to perform in preparation for a path sampling simulation.

The two-way shooting module and the “on-the-fly” flux/rate module both directly deal with how the simulation runs. Two-way shooting adds another option to the path sampling simulation, and the “on-the-fly” flux/rate adds a completely new type of simulation to *OPS*. In addition, the snapshot features module was a prerequisite for the two-way shooting module.

The path density, snapshot features, WHAM, and OPSPiggybacker modules focus on analysis. Path density plots and WHAM are standard tools in path sampling simulations, which were missing from *OPS*. The snapshot features module standardized access to several attributes commonly used in analysis, such that they are now independent of the underlying MD engine. The OPSPiggybacker module enables users to use the analysis tools of *OPS* with data that may have been generated by other, less robust software. In addition, the module for flux/rate from existing trajectories can be used either to help set up a simulation or as an analysis tool.

Finally, the network input module changed some of the internal data structures of *OPS*. This has the effect that information provided during setup is now available at analysis, without requiring it to be input again.

The next section provides more detailed descriptions of each of these modules.

3 Modules

Material in this section is largely drawn from the detailed module documentation files hosted at <http://e-cam-classical-md-modules.readthedocs.io/>. The “Motivation and exploitation” section describes the reason each module was developed, as well as providing specific examples of its use. Scientific details on the projects were provided in section 2.2. After the description of each module, links are provided to the specific module page in that documentation, as well as to the merge request where the module documentation was originally developed. Further details about the code contributed and the development process can be found through those links. In addition, each module consists of at least one example of showing how to use it, linked in the “Examples” section of the linked module documentation.

3.1 Path density

3.1.1 Module description

This module implements path density calculations for OpenPathSampling, including a generic multidimensional sparse histogram, and plotting functions for the two-dimensional case. Path density plots provide a way to visualize kinetic information obtained from path sampling, such as the mechanism of a rare event. In addition, the code in this module can also be used to visualize thermodynamic information such as free energy landscapes. This module has been incorporated into the core of OPS.

At first glance, a typical path density plot may appear similar to a two-dimensional free energy landscape plot. They are both “heatmap”-type plots, plotting a two-dimensional histogram in some pair of collective variables. However, path density differs from free energy in several important respects:

- A path density plot is histogrammed according to the number of paths, not the number of configurations. So if a cell is visited more than once during a path, it still only gets counted once.
- A path density plot may interpolate across cells that the path jumps over. This is because it is assumed that the input must actually be continuous.

These differences can prevent metastable regions from overwhelming the transition regions in the plot. When looking at mechanisms, the path density is a more useful tool than the raw configurational probability.

3.1.2 Motivation and exploitation

The path density is one of the most important tools for visualizing mechanisms, and is often one of the first things to analyze in order to draw scientific conclusions about the mechanism from transition path sampling simulations. Such a module was necessary to obtain useful results from TPS using OPS.

Practical exploitation:

- *KRas dynamics*: This module was used to illustrate the differences between dynamics of the wild-type and oncogenic mutant forms of KRas, as part of one student’s master’s thesis and another student’s bachelor’s thesis at the University of Amsterdam. Results from those projects are currently in preparation for publication [20].

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.2 Direct (on-the-fly) flux and rate calculation

3.2.1 Module description

This module contains code to implement the direct (on-the-fly) calculation of flux and rate in OpenPathSampling. This is especially useful for small (toy) systems, where you can easily run very long trajectories to get very accurate results, and would rather re-run than save the full trajectory. This module has been incorporated into the core of OPS.

Calculating the flux out of a state and through an innermost interface is one of the steps required in transition interface sampling (TIS). This module enables that. Although there are other ways to calculate the flux (for example, replica exchange TIS can calculate the flux as a byproduct of the “minus move”), frequently a direct calculation gives the best

statistics. In addition, an approximate flux should be calculated before running the simulation in order to ensure that the state and innermost interface definitions have been chosen well.

In principle, this module also makes it possible to calculate rates directly by running long molecular dynamics simulations. That won't be useful for very rare events, but it can be useful for validation.

3.2.2 Motivation and exploitation

An external calculation of the flux is required when doing multiple interface set TIS (MISTIS), and can be used in any situation. This direct approach makes it straightforward to calculate this rate within OPS.

Practical exploitation:

- *OPS introduction*: The MISTIS example for OPS required this module, and is in the paper introducing OPS [22].
- This module is also being used to validate the TIS simulations in OPS, in cases where the rate can be calculated with direct dynamics (including toy models and alanine dipeptide).

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.3 Improved input for OPS networks

3.3.1 Module description

This module includes modifications to OpenPathSampling that simplify the setup of transition networks, including providing a method so that extra information about interfaces can (optionally) be provided on setup in order to simplify analysis. This is primarily an infrastructure module, which will enable other modules for both simulation setup and analysis to be constructed more easily. This module has been incorporated into the core of OPS.

In OpenPathSampling, “reactions” (which are often conformational changes) are represented by objects that are called `Transitions`, and the set of all reactions of interest is represented by an object called a `TransitionNetwork`. The `TransitionNetwork` knows about all the reactions being sampled, as well as the path ensembles used to sample them. In general, a full reaction network might involve hundreds of path ensembles, so the `TransitionNetwork` is a factory that creates the ensembles so the user doesn't have to, and also provides conveniences for analysis, such as grouping the path ensembles according to the reaction they sample.

This module deals with two aspects of transition interface sampling methods. The first is the interface set, which is the group of interfaces associated with a given transition. These interfaces are associated with volumes in phase space. Those volumes are typically defined by a maximum value of some order parameter, λ . Knowing this edge value is essential for calculating rates.

The second aspect is the multiple state outer (MS-outer) interface ensemble. This is an ensemble used in some variants of multiple state transition interface sampling to facilitate replica exchange between paths with different initial states. In practice, this approach is likely to become less frequently used (there are more efficient approaches to achieve the same goals), so requiring that the MS-outer interface exist is not very forward-thinking, although removing entirely is also not the best approach.

Prior to the improvements made in this module, OpenPathSampling suffered from the following problems:

- An interface had no way of knowing what its “edge” value was, only whether a given snapshot was inside it or outside it. This made it difficult to automatically determine the value at the outermost interface for calculating the rate. The previous code relied on a trick that assumed that trajectories in the interface had a relatively low probability of crossing to another state. In addition, other approaches (such as interface optimization approaches) can benefit from knowing this value for all interfaces, and from having the ability to create a new interface from a single value of the order parameter λ .
- All networks required a multiple state outer interface, even if it wasn't used. This also meant that the outermost interface a user defined was converted to an MS-outer interface, which could lead to confusion. This module makes usage much easier to understand.

This module changes the setup of interface sets, such that they can identify their edge values (if it is uniquely identified; the code still works if it is not unique). This obviates the need for a trick (that wasn't guaranteed to work) to guess where the boundary was.

This module also changes the way that the multiple state outer interface is set up. Now the user must explicitly make a multiple state outer interface object, which will then make the appropriate MS-outer ensemble. We wanted to keep the ability to have an MS-outer ensemble. However, we did not want to require it, because there are, in general, better approaches to accomplish the same things.

3.3.2 Motivation and exploitation

This module has two primary purposes: to simplify usage of existing tools in OPS, and to provide the infrastructure that makes it possible for users to develop new tools.

Practical exploitations:

- *OPS introduction*: This module is used in the setup of all TIS examples in the paper introducing OPS [22].
- The module for interface optimization, implemented during the ESDW in Traunkirchen and to be delivered as part of D1.3, required development of this module.

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.4 New WHAM code for OPS

3.4.1 Module description

This module includes a re-write of the OpenPathSampling Weighted Histogram Analysis Method (WHAM) code, used for rate calculations with transition interface sampling. This fixes limitations of the previous version and is more readable. This module has been incorporated into the core of OPS.

Methods like TIS sample multiple ensembles, and then combine the results from the individual restricted ensembles to obtain results for the unrestricted (natural) ensemble, such as the rates of a reaction or projections of the free energy surface. One approach to combining these ensembles is WHAM. This module provides an implementation of WHAM that is specialized for path sampling. Details about the WHAM method (as used for calculating free energies) can be found in Frenkel and Smit, section 7.3 [23].

The module is a rewrite of previous code in OPS. The previous code had several limitations, most notably the assumption that all ensembles had the same number of sampling. Practical cases required that the number of samples in each ensemble be allowed to vary. In addition, the previous code was poorly documented and untested. This module fixes all of that, and includes detailed comments connecting the code to the equations in Frenkel and Smit.

3.4.2 Motivation and exploitation

This code was re-written because a practical exploitation of the existing code required the ability to have different numbers of samples in each ensemble. The original code was poorly documented and untested, and therefore difficult to maintain and update, so it was rewritten to be easier to maintain, while also adding the ability to have different numbers of samples per ensemble.

Practical exploitations:

- *WC-HG DNA*: This module was used to calculate the rate of the conversion between the Watson-Crick and Hoogsteen base pairing motifs in DNA [17] from TIS data generated by a legacy script.
- *OPS introduction*: This module is used for the analysis of all the TIS examples in the paper introducing OPS [22].

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.5 Flux/Rate from existing trajectories

3.5.1 Module description

This module adds the ability to use existing trajectories to calculate the flux through an interface or the rate of a transition in OpenPathSampling. This differs from the on-the-fly flux/rate module in that it requires trajectories that have already been generated. This module has been incorporated into the core of OPS.

Calculating the flux out of a state and through a given interface is an important step in transition interface sampling (TIS). It is required for the rate calculation, and can be used to determine the best location for the innermost interface before performing the main TIS sampling.

In many cases, the main TIS sampling is only performed after a significant amount of direct MD has been performed. For example, MD trajectories may have already been used to identify and test the stability of stable states. This module calculates the flux (as well as the rate, a related quantity) from existing trajectories. These can be trajectories generated with OPS or loaded into OPS from other simulation packages, such as Gromacs.

The flux calculation is used as part of obtaining the rate in TIS. While some variants of TIS calculate the flux as part of their primary sampling, others need an auxiliary calculation. In addition, the flux calculation is always useful for confirming a good definition of the innermost interface in TIS, and using an existing trajectory to select the location of the innermost interface does not require re-running the dynamics.

The rate calculation will probably not be used very much, because rates require extremely long trajectories. Both flux and rate are included as a single “module” because the code is very closely related.

3.5.2 Motivation and exploitation

This module allows us to reuse work from existing MD simulations for the definition of TIS interfaces. In particular, this module was needed by two scientific projects; both of which are used as examples for the module (see the module documentation link below) as well as practical exploitations.

Practical exploitations:

- *WC-HG DNA*: This module was used to set the location of the innermost interfaces for the TIS simulations of the WC-HG transition, as well as to calculate the flux through those innermost interfaces. This was essential to calculating the rates in Ref. [17].
- *DFG flip in kinases*: Using data generated from Folding@Home, this module was used to test proposed state definitions for the DFG flip in Src and Abl. It made it very easy to identify the transitions for a proposed pair of state definitions, and to identify frames from the associated trajectory.

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.6 OPS snapshot features

3.6.1 Module description

This module includes several new OpenPathSampling snapshot “features,” which make attributes directly accessible from the snapshot object. In particular, this includes support for features to access `masses`, `n_degrees_of_freedom`, and `instantaneous_temperature` in both the toy and OpenMM engines. This module has been incorporated into the core of OPS.

In OpenPathSampling, certain quantities can be accessed directly from each snapshot. The standard examples of such “features” are things like coordinates and velocities, which are stored for each snapshot. However, additional features can also be added, which may not require per-snapshot storage. This approach makes them accessible from the snapshot as `snapshot.feature`, just like a stored quantity, even if they are not stored. For example, the `snapshot.masses` can actually be a pointer to a single array of masses for all snapshots from a given engine, and `snapshot.instantaneous_temperature` can actually be a quantity that is computed on demand, rather than stored. This module makes several new features available for snapshots from both the OpenMM engine and from the toy engine.

By adding a standard interface for snapshots to provide similar information, this provides several advantages:

- Many analysis tools require the masses. This module makes it easier to apply analysis tools for one MD engine to the results from another MD engine.
- Other modules will require a standardized interface. In particular, the two-way shooting module will require both the masses and the number of degrees of freedom, and future modules for collective variables are also likely to need to masses.
- This provides examples of how to implement snapshot “features,” which are necessary for the implementation of new engine modules.
- The instantaneous temperature, in particular, is an important check that a simulation has been well-behaved. Drift of the instantaneous temperature is a sign of a problem in the simulation.

3.6.2 Motivation and exploitation

This is primarily an infrastructure module, which enables other modules to be built on top of it. It also simplifies the code for specific applications, which would have previously required different approaches depending on the underlying MD engine.

Practical exploitations:

- The module for two-way shooting, also delivered in this report, required access to both the masses and the number of degrees of freedom, which are implemented in this module.

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.7 Two-way shooting

3.7.1 Module description

This module provides the ability to study rare events with two-way shooting in OpenPathSampling. This module has been incorporated into the core of OPS.

Different types of dynamics are better suited for different kinds of Monte Carlo moves in path sampling. “One-way shooting,” which was already implemented in OpenPathSampling, is an efficient approach for sampling paths when the dynamics are chaotic or diffusive. However, it requires stochastic dynamics, and therefore isn’t appropriate for deterministic dynamics, as should be used with ballistic processes. For ballistic processes and deterministic dynamics, the “two-way shooting” move, which is implemented in this module, should be used. These moves differ in that one-way shooting selects a frame as a shooting point and evolves the trajectory *either* forward or backward, keeping part of the original trajectory, whereas two-way shooting selects a shooting point, modifies it (usually by changing the velocities), and evolves *both* forward and backward. In one-way shooting, the stochastic dynamics obviates the need to modify the shooting point.

The shooting point selection methods used by one-way shooting can be re-used for two-way shooting. However, two-way shooting requires a `SnapshotModifier`, which one-way shooting does not. The basics of the `SnapshotModifier`, as well as an implementation which completely randomizes the velocities according to the Boltzmann distribution, were included in a module to do committor simulations.

3.7.2 Motivation and exploitation

As one of the original sampling moves developed for TPS, implementing two-way shooting was an important part of E-CAM’s stated goal of developing a library of modules for path sampling. In addition, two-way shooting can be a component of other modules in this library. Two-way shooting was explicitly mentioned as a module for E-CAM development in section 4.2 of Deliverable D1.1, and falls in the category of “specific modules that perform [path sampling simulations]” in the community needs section of report from the State of the Art Workshop.

Practical exploitations:

- One of the targets for the ESDW in Traunkirchen is the aimless shooting algorithm, which is a variant of two-way shooting. This module will be used to develop the aimless shooting module for the ESDW.

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.8 Annotated trajectories

3.8.1 Module description

When dealing with biomolecular systems, one of the common challenges is to define the (meta)stable states. The existence of metastable states is easily determined when visually inspecting the trajectory. However, identifying geometric criteria to characterize the states remains challenging.

This module provides a data structure that allows the user to easily annotate a trajectory with the visually identified states, and to compare those annotations to proposed state definitions. It also includes tools to visualize where the proposed state definition matches the annotations.

3.8.2 Motivation and exploitation

This module was motivated by challenges in identifying metastable states, in particular for the dynamics of flexible protein regions. It will also be useful for creating automated methods for identifying states, including some approaches that are currently under development. As a state definition assessor, this module meets one of the community needs identified in section 4 of the State of the Art Workshop report.

Practical exploitations:

- *KRas dynamics*: This module was specifically inspired by challenges in identifying states in the KRas system. The transitions of interest in KRas involve flexible regions with multiple metastable intermediates. Although this module was not used for the KRas system, its purpose was to make it easier to solve the same problems that were encountered there, and KRas was used as a check for the usability of this module.

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

3.9 OPS Piggybacker

3.9.1 Module description

This module provides a library for converting path sampling simulations from legacy codes into a format that can be analyzed by OpenPathSampling. The implementation in this module includes some basic path sampling techniques; later development could extend this to include more simulation types.

OpenPathSampling contains excellent tools for analyzing simulations, as well as excellent tools for sampling. However, since OpenPathSampling is a new software package, users may already have simulations that they have run with other packages. The purpose of this module is to provide tools that allow the user to easily convert legacy script output into a format that can be analyzed by OpenPathSampling. It is a library of tools that can be used with data from any existing path sampling simulation output, and specifically includes tools to simplify the use of flexible-length TPS with one-way shooting and uniform shooting point selection. Extending to the library to work with other simulation types will be part of future work.

The OPSiggybacker essentially fakes a simulation, based on data from another source. In this module, it has the ability to read in data from one-way TPS. The user must create the appropriate OPS `TransitionNetwork` object (including defining the correct collective variables and state volumes). Then the code creates a `MoveScheme`, but instead of actually running the simulation, it reads in the results of an existing simulation and provides the same output that the `MoveScheme` *would* have provided. We call this a “pseudo-simulation.”

3.9.2 Motivation and exploitation

One of the most important steps in helping users benefit from the E-CAM path sampling library, and in particular its analysis tools, is to make it possible to them to use previously existing simulation data. Since there is no widely-

used tool for path sampling, there is also no established format for path sampling simulation data. Instead, individual research groups develop their own format as they develop their own code. Because of this, it is essential that a generic library be developed to facilitate the translation of that existing data to formats that can be analyzed by the tools in development by E-CAM. This module provides that library.

Practical exploitations:

- *WC-HG DNA*: This module was used to enable use of OPS-based modules to analyze the data from the WC-HG DNA project, which was generated with an older, in-house script. Missing parts of the data from the original simulation prevented that analysis from being used in publication, but the attempt proved the viability and usefulness of this module.

Additional information:

- [Module documentation](#)
- [Documentation merge request](#)

4 Performance Considerations

In section 2.3, we discussed the three stages of running a simulation: setting it up, running it, and analyzing the results. The modules delivered here include contributions to all three stages. However, the vast majority of computational time is spent in the second stage, generating the simulation data (specifically, in performing molecular dynamics). Therefore, discussion of performance primarily focuses on the modules associated with that stage.

In [E-CAM Deliverable D1.1](#) [1], we discussed the variety of MD engines. As mentioned in that report, different communities have coalesced around different engines. Software for trajectory-based rare events simulations, such as OpenPathSampling, usually wraps around other MD engines, because these simulations don't change the underlying dynamics. This gives them the potential for a wider range of applications. Furthermore, since the primary cost of the calculation is in performing the dynamics, this also means that most of the responsibility for performance falls on the underlying engine. OPS already has support for OpenMM, a GPU-accelerated MD code, and support is in development for the widely-used and highly-scalable packages LAMMPS and Gromacs. Each of these dynamics engines is designed for high performance computing, and in this way, OPS inherits the performance from the underlying dynamics code.

The modules for “direct (on-the-fly) flux and rate calculation” and for “two-way shooting” both wrap around such dynamics engines, and their performance characteristics therefore depend on the underlying engine. The objects created by both of these modules take an instance of the OPS DynamicsEngine class in their initialization. The DynamicsEngine abstracts specifics of the underlying engine so that the same code in these modules can be used to manage different underlying engines.

This approach means that as new generations of hardware lead to new molecular dynamics codes or changes in existing codes, OpenPathSampling, and the modules designed for it, can continue to use the cutting edge in hardware with minimal modification.

5 Outlook

The report of Deliverable 1.2 of E-CAM describes 9 software modules in classical dynamics. They include codes and documentation for tools for sampling trajectories in order to study the thermodynamics and kinetics of rare events. These modules interface with, and many are included in, the open-source package OpenPathSampling, and they have been accepted into the E-CAM software library.

The modules that have been included cover all stages in a typical path sampling simulation: simulation setup, running the simulation, and analyzing the results. As with other parts of the OPS package, the simulation performance is inherited from the underlying MD engine, enabling these modules to automatically use the underlying engine's support for new hardware developments.

These modules have a wide range of practical applications, and are already being used to determine the mechanism and rates of the conformational changes between Watson-Crick and Hoogsteen base pairing motifs in DNA, to understand the difference between the dynamics of wild-type and oncogenic forms of KRas, and to analyze data about the DFG flip in kinases, a process related to inhibitor binding. In addition, several of these modules provide basic examples of common path sampling tools in the OPS package. Two of these modules, two-way shooting and path density plots, were explicitly listed as targets for E-CAM development in Deliverable D1.1 [1], and another module, annotated trajectories, meets one of the community needs listed in the State of the Art Workshop report⁴. The modules delivered here respond to the real needs of the classical MD community, and are of practical use for exploitation by academia and industry.

WP1 held its first ESDW in Traunkirchen, Austria, during the second half of November 2016. The 6 modules from that ESDW, which were not completed in time for the January 2017 deadline for this report, were:

- Reactive flux
- Transition state ensemble
- Interface optimization
- Reaction coordinate from maximum likelihood
- Aimless shooting
- Shifting move

Most of those modules — all except aimless shooting and the shifting move — deal with topics at the intersection of the focus of this report (modules for trajectory sampling and the thermodynamics and kinetics of rare events) and the focus of Deliverable D1.3 (modules for statistical and machine learning tools for the analysis of rare events). Therefore, those modules will be included in Deliverable D1.3.

Some of the modules from the ESDW required the development of modules in this report. For example, aimless shooting is a variant of two-way shooting, and the interface optimization algorithm requires the tools created in the module for improved input for OPS networks. In addition, many of the ESDW modules were specifically mentioned as targets for development, either in D1.1 or on the State of the Art Workshop report. As such, future modules from E-CAM Work Package 1 will build on the modules delivered here, and will continue to respond to the needs that E-CAM has identified within the classical MD community.

⁴Report from WP1 State of the Art Workshop available for download from the E-CAM website: <https://www.e-cam2020.eu/scientific-reports/>

References

Acronyms Used

CECAM Centre Européen de Calcul Atomique et Moléculaire

ESDW Extended Software Development Workshop

HG Hoogsteen

MD molecular dynamics

OPS OpenPathSampling

TPS transition path sampling

TIS transition interface sampling

WC Watson-Crick

WHAM Weighted Histogram Analysis Method

PDRA Postdoctoral Research Associate

URLs referenced

Page ii

<https://www.e-cam2020.eu> ... <https://www.e-cam2020.eu>

<https://www.e-cam2020.eu/deliverables> ... <https://www.e-cam2020.eu/deliverables>

Internal Project Management Link ... <https://redmine.e-cam2020.eu/issues/130>

dwhs@hyperblazer.net ... <mailto:dwhs@hyperblazer.net>

<http://creativecommons.org/licenses/by/4.0> ... <http://creativecommons.org/licenses/by/4.0>

Page 1

E-CAM Deliverable D1.1 ... <https://dx.doi.org/10.5281/zenodo.841694>

report of the E-CAM Classical MD State of the Art Workshop ... <https://www.e-cam2020.eu/wp-content/uploads/2017/02/SAW-WP1.pdf>

<https://www.e-cam2020.eu/scientific-reports/> ... <https://www.e-cam2020.eu/scientific-reports/>

OPS ... <https://openpathsampling.org>

Jupyter notebook ... <http://jupyter.org>

Page 2

<http://e-cam-classical-md-modules.readthedocs.io> ... <http://e-cam-classical-md-modules.readthedocs.io>

<https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules> ... <https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules>

E-CAM Deliverable D1.1 ... <https://dx.doi.org/10.5281/zenodo.841694>

<https://www.e-cam2020.eu/scientific-reports/> ... <https://www.e-cam2020.eu/scientific-reports/>

OpenPathSampling GitHub repository ... <http://github.com/openpathsampling/openpathsampling>

annotated trajectories module ... https://github.com/dwhswenson/annotated_trajectories

OPSPiggybacker module ... <https://github.com/dwhswenson/OPSPiggybacker>

ops_additional_examples repository ... https://gitlab.e-cam2020.eu/dwhswenson/ops_additional_examples

Page 4

millisecond dynamics of a protein ... <http://pubs.acs.org/doi/abs/10.1021/acs.jpcc.6b02024>

TPS ... <http://aip.scitation.org/doi/abs/10.1063/1.475562>

TIS ... <http://aip.scitation.org/doi/abs/10.1063/1.1562614>

2010 review by Bolhuis and Dellago in Reviews in Computational Chemistry ... <http://onlinelibrary.wiley.com/doi/10.1002/9780470890905.ch3/summary>

2008 review by Dellago and Bolhuis in Advances in Polymer Science ... https://link.springer.com/chapter/10.1007%2F978-3-540-87706-6_3

Wikipedia entry on path sampling ... https://en.wikipedia.org/wiki/Transition_path_sampling

Aaron Keys's tutorial on path sampling ... <https://sites.google.com/site/aaronskeys/resources/tutorials/transition-path-sampling>

Mechanisms of complex chemical reactions, such as autoionization of water ... <http://science.sciencemag>.

org/content/291/5511/2121

Mechanism of hydrophobic assembly ... <http://pubs.acs.org/doi/abs/10.1021/jp077186+>

Evidence that the glass transition is a first-order phase transition ... <http://science.sciencemag.org/content/323/5919/1309>

Mechanism of crystal nucleation ... <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.106.085701>

Mechanism of cavitation ... <http://www.pnas.org/content/113/48/13582>

Identifying new mechanisms in catalytic systems ... <http://www.sciencedirect.com/science/article/pii/S0920586105002099>

Characterization of the conformational dynamics networks in proteins ... <http://www.pnas.org/content/107/6/2397>

Page 7

<http://e-cam-classical-md-modules.readthedocs.io/> ... <http://e-cam-classical-md-modules.readthedocs.io/>

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_path_density/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/1

Page 8

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_direct_rate_flux/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/7

Page 9

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_network_input/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/3

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_new_wham/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/5

Page 10

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_flux_rate_analysis/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/4

Page 11

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_snapshot_features_1/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/19

Page 12

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_two_way_shooting/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/21

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/annotated_trajectories/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/18

Page 13

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/ops_piggybacker/readme.html

Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/22

Page 14

E-CAM Deliverable D1.1 ... <https://dx.doi.org/10.5281/zenodo.841694>

Page 15

Deliverable D1.1 ... <https://dx.doi.org/10.5281/zenodo.841694>

<https://www.e-cam2020.eu/scientific-reports/> ... <https://www.e-cam2020.eu/scientific-reports/>

-
- [1] Christoph Dellago, David Swenson, Donal MacKernan, Ralf Everaers, and Jony Castanga. Identification/selection of E-CAM MD codes for development, November 2016. URL <https://doi.org/10.5281/zenodo.841694>.
 - [2] Kresten Lindorff-Larsen, Paul Maragakis, Stefano Piana, and David E. Shaw. Picosecond to millisecond structural dynamics in human ubiquitin. *J. Phys. Chem. B*, 120(33):8313, 2016.
 - [3] Christoph Dellago, Peter G Bolhuis, Félix S Csajka, and David Chandler. Transition path sampling and the calculation of rate constants. *J. Chem. Phys.*, 108(5):1964, 1998.
 - [4] Titus S van Erp, Daniele Moroni, and Peter G Bolhuis. A novel path sampling method for the calculation of rate constants. *J. Chem. Phys.*, 118(17):7762, 2003.
 - [5] Peter G Bolhuis and Christoph Dellago. Trajectory-Based Rare Event Simulations. In *Reviews in Computational Chemistry, Volume 27*, pages 111–210. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2010.
 - [6] Christoph Dellago and Peter G Bolhuis. Transition Path Sampling and Other Advanced Simulation Techniques for Rare Events. In *Advanced Computer Simulation Approaches for Soft Matter Sciences III*, pages 167–233. Springer, Berlin, Heidelberg, Berlin, Heidelberg, 2008.
 - [7] Wikipedia. Transition path sampling — Wikipedia, the free encyclopedia, 2017. URL https://en.wikipedia.org/w/index.php?title=Transition_path_sampling&oldid=720483427. [Online; accessed 20-November-2017].
 - [8] Aaron Keys. Transition Path Sampling, 2017. URL <https://sites.google.com/site/aaronskeys/resources/tutorials/transition-path-sampling>. [Online; accessed 20-November-2017].
 - [9] Phillip L Geissler, Christoph Dellago, David Chandler, Jürg Hutter, and Michele Parrinello. Autoionization in Liquid Water. *Science*, 291(5511):2121, 2001.
 - [10] Adam P Willard and David Chandler. The role of solvent fluctuations in hydrophobic assembly. *J. Phys. Chem. B*, 112(19):6187, 2008.
 - [11] L O Hedges, R L Jack, J P Garrahan, and D Chandler. Dynamic Order-Disorder in Atomistic Models of Structural Glass Formers. *Science*, 323(5919):1309–1313, 2009.
 - [12] Wolfgang Lechner, Christoph Dellago, and Peter G Bolhuis. Role of the Prestructured Surface Cloud in Crystal Nucleation. *Phys. Rev. Lett.*, 106(8):085701, 2011.
 - [13] Georg Menzl, Miguel A Gonzalez, Philipp Geiger, Frédéric Caupin, José L F Abascal, Chantal Valeriani, and Christoph Dellago. Molecular mechanism for cavitation in water under tension. *PNAS*, 113(48):13582, 2016.
 - [14] Cynthia S Lo, Ravi Radhakrishnan, and Bernhardt L Trout. Application of transition path sampling methods in catalysis: A new mechanism for CC bond formation in the methanol coupling reaction in chabazite. *Catalysis Today*, 105(1):93, 2005.
 - [15] Jocelyne Vreede, Jarek Juraszek, and Peter G Bolhuis. Predicting the reaction coordinates of millisecond light-induced conformational changes in photoactive yellow protein. *Proc. Natl. Acad. Sci. U.S.A.*, 107(6):2397, 2010.
 - [16] Evgenia N Nikolova, Ioan Andricioaei, Hashim M Al-Hashimi, Eunae Kim, Abigail A Wise, and Patrick J O'Brien. Transient Hoogsteen base pairs in canonical duplex DNA. *Nature*, 470(7335):498, 2011.
 - [17] Jocelyne Vreede, Peter G. Bolhuis, and David W.H. Swenson. Path sampling simulations of the Watson-Crick to Hoogsteen conversion in DNA. In prep.
 - [18] Ken S Lau and Kevin M Haigis. Non-redundancy within the RAS oncogene family: Insights into mutational disparities in cancer. *Mol Cells*, 28(4):315, 2009.
 - [19] M Spoerner, C Herrmann, I R Vetter, H R Kalbitzer, and A Wittinghofer. Dynamic properties of the Ras switch I region and its importance for binding to effectors. *Proc. Natl. Acad. Sci. U.S.A.*, 98(9):4944, 2001.

-
- [20] Sander Roet, Ferry Hooft, Peter G. Bolhuis, David W.H. Swenson, and Jocelyne Vreede. Simulating the dynamics of oncogenic and wild-type KRas. In prep.
- [21] Jianming Zhang, Priscilla L Yang, and Nathanael S Gray. Targeting cancer with small molecule kinase inhibitors. *Nat Rev Cancer*, 9(1):28, 2009.
- [22] Jan-Hendrik Prinz, David W.H. Swenson, Peter G. Bolhuis, and John D. Chodera. OpenPathSampling: A Python framework for path sampling simulations. I. Introduction and usage. In prep.
- [23] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, San Diego, 2nd edition, 2002.