



Classical MD E-CAM Modules II

E-CAM Deliverable 1.3

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	D1.3
Deliverable Nature	Report
Dissemination Level	Public
Contractual Date of Delivery	Project Month 24 (1 st October, 2017)
Actual Date of Delivery	12 th December, 2017
Description of Deliverable	9 software modules in classical Molecular Dynamics delivered to the E-CAM repository in the area of statistical and machine learning tools for the analysis of rare events, and their documentation.

Document Control Information

Document	Title:	Classical MD E-CAM Modules II
	ID:	D1.3
	Version:	As of 12 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:	Jony Castagna (STFC, Sci-Tech Daresbury Laboratory), Raffaella Cabriolu (Norwegian University of Science and Technology), Anastasiia Maslechko (Norwegian University of Science and Technology) Clemens Moritz (University of Vienna), Sander Roet (University of Amsterdam), Andreas Singraber (University of Vienna), Christoph Dellago (University of Vienna)
	Reviewed by:	Sara Bonella (EPFL)
	Approved by:	Christoph Dellago (University of Vienna)

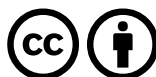
Document Keywords

Keywords:	E-CAM, Molecular Dynamics, Module, Rare Events, Path Sampling, Open-PathSampling, Machine Learning, Statistical Analysis, CECAM
-----------	---

12th 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 Jony Castagna (STFC, Sci-Tech Daresbury Laboratory), Raffaella Cabriolu (Norwegian University of Science and Technology), Anastasiia Maslechko (Norwegian University of Science and Technology) Clemens Moritz (University of Vienna), Sander Roet (University of Amsterdam), Andreas Singraber (University of Vienna), Christoph Dellago (University of Vienna). 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	6
3.1 Committor analysis	6
3.1.1 Module description	6
3.1.2 Motivation and exploitation	6
3.2 Reactive flux	7
3.2.1 Module description	7
3.2.2 Motivation and exploitation	7
3.3 S-Shooting algorithm	7
3.3.1 Module description	7
3.3.2 Motivation and exploitation	7
3.4 Transition state ensemble	7
3.4.1 Module description	8
3.4.2 Motivation and exploitation	8
3.5 New TIS analysis framework	8
3.5.1 Module description	8
3.5.2 Motivation and exploitation	9
3.6 Interface optimization	9
3.6.1 Module description	9
3.6.2 Motivation and exploitation	9
3.7 Maximum likelihood for reaction coordinates	9
3.7.1 Module description	10
3.7.2 Motivation and exploitation	10
3.8 Channel analysis	10
3.8.1 Module description	10
3.8.2 Motivation and exploitation	11
3.9 Resampling statistics	11
3.9.1 Module description	11
3.9.2 Motivation and exploitation	11
4 Performance Considerations	12
5 Outlook	13
References	14

Executive Summary

In this report for Deliverable 1.3 of E-CAM, 9 software modules in classical dynamics are presented. These modules represent improvements and new features in software for machine learning and statistical analysis of rare events. As such, they primarily focus on the analysis stage of simulation, although some modules also affect the way data is generated to study the rare event.

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 Workshop](#)², 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 enable junior participants at the Extended Software Development Workshops to contribute modules, which required a careful selection of proposed modules based on the feasibility of accomplishing them in the context of the workshop.
- 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.

The 9 modules presented here are:

- Committor analysis
- Reactive flux
- S-Shooting algorithm
- Transition state ensemble
- Interface optimization
- Maximum likelihood for reaction coordinates
- Channel analysis
- Resampling statistics
- New transition interface sampling (TIS) analysis framework

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 develops 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, as well as from previous deliverables.

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 second group of nine modules delivered as part of E-CAM WP1. As described in the grant agreement, these modules are “in the area of statistical and machine learning tools for the analysis of rare events.” Of these modules, 4 were developed by the E-CAM WP1 Postdoctoral Research Associate (PDRA), 4 were developed at the first WP1 Extended Software Development Workshop (ESDW), held in Traunkirchen in November 2016³, and 1 was developed at the second WP1 ESDW, held in Leiden in August 2017⁴. Two other modules were completed at the Traunkirchen ESDW, but do not fit the topic of this deliverable, and at least 5 other modules are in development based on work done during the Leiden ESDW.

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 a committor analysis and the reactive flux method, which are included as part of this Deliverable. The reactive flux module is also specifically mentioned as a target for E-CAM development in section 4 of the report⁵ from the State of the Art Workshop held in Leiden in 2016. Additionally, that report specifically mentions the need for a transition state ensemble module and an interface optimization module, which are both delivered in this report. It also mentions a need for reaction coordinate analysis modules, of which the “maximum likelihood for reaction coordinates” module is an example, and for general “analysis tools to work with path sampling,” which includes many of the other modules, such as the new TIS analysis framework, the channel analysis module, and the resampling statistics module.

This report includes several modules that were contributed from ESDWs. The ESDW format provides a period of less than two weeks during which as much of the software development as possible should be done. This requires a careful selection of the modules to be implemented — it must be feasible to complete the core development of any project we propose within that time period. That limits the complexity of those modules. Selection of modules for the ESDW was

³CECAM Flagship workshop in 2016: <https://www.cecam.org/workshop-0-1356.html>

⁴CECAM Flagship workshop in 2017: <https://www.cecam.org/workshop-0-1406.html>

⁵Report available for download from the E-CAM website: <https://www.e-cam2020.eu/scientific-reports/>

driven by the combination of feasibility and relevance to the goals of the Deliverable and of the project, as laid out in previous reports. They were not selected based on the immediate needs of a specific scientific project, and therefore have fewer practical exploitations so far. However, these software development tasks were used as part of a practical introduction to advanced programming techniques and hardware environments for the participants of the ESDWs. Therefore, they have high value for the training component of E-CAM.

The PDRA-developed modules have been merged into the core of OPS, and therefore the code is stored as pull requests on the official [OpenPathSampling GitHub repository](#). The ESDW-developed modules are hosted in repositories on the E-CAM GitLab service, although several of them are likely to be accepted as contributions to the core OPS code in the near future. Additional examples for several modules are also hosted on E-CAM's GitLab service, 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 rare events, and we will provide 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 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 [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 ESDWs. 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 main 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].

- **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 [21].

2.3 The role of these modules in the path sampling workflow

The process of running a simulation can be split into three stages:

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

Since the topic of this report is modules in the area of machine learning and statistical tools for the analysis of rare events, the modules included in this report primarily focus on analysis. In particular, the maximum likelihood for reaction coordinates, channel analysis, resampling statistics, and new TIS analysis framework modules are very focused on analysis.

Another group of modules provide new ways of generating trajectories, along with the appropriate analysis tools. These are the committor analysis, the reactive flux, and the S-Shooting modules. Although they build on the OpenPathSampling code, these are not, strictly speaking, path sampling methods. As completely new simulation types, they are involved in all three stages of the simulation process. However, they use much of the OPS machinery to simplify the development of other simulations for the study of rare events.

The remaining modules wrap around the outside of the simulation process. They take the results from one simulation, and analyze them in order to decide how to run a new simulation. These are the transition state ensemble and interface optimization modules.

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 for more information about the modules. All modules have a link to the merge request where the module documentation was developed. Modules that have been completed and accepted into the E-CAM library (all except S-shooting) also have a link to the specific module page in the E-CAM library documentation website. 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 Committor analysis

This module adds a simulator to perform committor analysis in OpenPathSampling, given a set of initial points to shoot from. The committor analysis module itself is a statistical analysis tool, however, it is also a prerequisite for one of the successful applications of machine learning to the analysis of rare events: the use of neural networks to predict committor isosurfaces [22].

3.1.1 Module description

A reaction coordinate is a function of the coordinates (and rarely of the momenta) of the atoms of the system whose values provide a good characterization of the process of a rare event. A trivial example might be the distance between a ligand and its binding site in a protein in the simulation of a docking experiment. Among these functions, a special role is played by the committor.

The committor for a given configuration (in the context of some transition $A \rightarrow B$) is defined as $p_B(x)$, the probability that a trajectory beginning at configuration x will reach state B before state A . The isosurfaces of the committor are a good definition of the reaction coordinate (the probability of ending in the product state is certainly a measure of the progress of the reaction). The transition state will have an equal chance of going to either state, so configurations with a committor of approximately 50% are said to make up the “transition state ensemble.” As a result, a committor simulation is essential both for the definition of the reaction coordinate and for the verification of a proposed transition state. This module provides a straightforward way of calculating the committor for a given set of initial conditions.

In addition to calculating the committor, this module can be used to generate initial trajectories for transition path sampling. Initial configurations can come from unphysical processes (such as biased dynamics), and the committor calculation can be used to generate trajectory segments that link a single configuration to each of the final states. Such segments could be joined to make an initial path which is not entirely physical (due to the kink in the velocities from random velocity selection in the committor), but that will usually be removed after a short equilibration with path sampling.

3.1.2 Motivation and exploitation

The committor was explicitly identified as a target for E-CAM development in D1.1. It is an important tool in the validation of a proposed transition state. In addition, much of the code of the committor simulation could be refactored in order to be reused by both reactive flux and S-shooting, making those modules much easier for ESDW participants to develop. Additionally, the module to identify the transition state ensemble wraps around the committor analysis, and therefore also required it.

Practical exploitation:

- *KRas dynamics*: The committor analysis was used to generate initial paths that included velocities, based on input trajectories that didn't, for some of the KRas TPS simulations. A publication from that project is in preparation [20].

Additional information:

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

3.2 Reactive flux

3.2.1 Module description

The reactive flux method in combination with a free energy calculation allows to derive the rate constant of a rare event. This is accomplished by a shooting algorithm similar to a committor analysis where fleeting trajectories starting from the dividing surface are generated and statistics about their state with respect to a collective variable is collected. There are many flavors of the reactive flux method, this module implements the effective positive flux method as described by van Erp and Bolhuis [23].

3.2.2 Motivation and exploitation

Reactive flux was identified by both Deliverable D1.1 and the State of the Art Workshop report as a target for E-CAM development. Since this module could be based on the committor analysis, it was set as a target for the first WP1 ESDW in Traunkirchen. Development was driven by the broad community need and the feasibility as a target for training at an ESDW, as opposed to a specific scientific project. Therefore, it has not had any direct practical exploitations yet. However, it is also a prerequisite for the S-shooting module, which was developed at the second WP1 ESDW.

Additional information:

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

3.3 S-Shooting algorithm

This module builds on OpenPathSampling to implement the S-shooting algorithm, a recently developed method similar to reactive flux to calculate rates.

3.3.1 Module description

S-shooting [24] is a recently developed method to determine rate constants of rare events. It is similar in spirit to the reactive flux method but its relaxed requirements help to overcome practical problems in the reactive flux method. The method is based on a simple shooting algorithm where trajectories are propagated forward and backward in time for a fixed number of timesteps. The starting points need to be provided and must lie in the saddle point region. This so-called S region (hence the name S-shooting) is defined via a suitable reaction coordinate and is required to separate the stable states A and B in such a way that no trajectory can connect A with B without visiting S. In contrast to the reactive flux method the time derivative of the reaction coordinate is not required which makes this approach applicable to systems exhibiting diffusive dynamics along the reaction coordinate. The S-shooting method can also be applied if the initial shooting points are taken from a biased simulation. Thus, it is a natural follow-up to free energy calculations like umbrella sampling and in combination with free energy curves allows the computation of rate constants.

3.3.2 Motivation and exploitation

This module was developed as part of the second WP1 ESDW, held in Leiden in August 2017, and was selected based on a participant's specific interest and experience with this method. As this is a very recently developed module, it has no practical exploitation yet, but it is an excellent example of using an ESDW participant's previous experience to meet E-CAM's goal of training through module development.

Additional information:

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

3.4 Transition state ensemble

This module is an addition to OpenPathSampling to calculate configurations that correspond to the transition state ensemble from a list of trajectories.

3.4.1 Module description

Often in transition path sampling we want to get an idea about the features of the transition state. This is done by generating an ensemble of configurations that correspond to a committor of approximately 50%. This ensemble gives information about the transition state and the shape of the barrier. This code provides a straightforward way of calculating this ensemble for a given list of trajectories.

This module searches for a single transition state frame from each trajectory. This is done by bisection of the trajectory, depending on the current committor. For example, if the current committor is too high (too much ends up in state B) the next index is selected halfway towards the left edge and the current index is set as the new right edge. This is repeated until a committor within a given range is reached or no new frame can be selected.

Whereas the committor module can tell whether a given snapshot is in the transition state ensemble, this module implements an efficient search for such snapshots.

3.4.2 Motivation and exploitation

The report from the E-CAM Molecular Dynamics State of the Art Workshop, held in Leiden in 2016, specifically mentions calculation of the transition state ensemble as a target for E-CAM development. Once the committor analysis module (which this module builds on) was written, this was identified as a module to be developed at an ESDW as part of training goals of E-CAM.

Practical exploitation:

- *KRas dynamics*: An early version of this module was used to generate initial trajectories for TPS based on input trajectories that did not have associated velocities. The publication from that project is in preparation [20].

Additional information:

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

3.5 New TIS analysis framework

This module provides a new framework for analyzing transition interface sampling simulations using OpenPathSampling. The previous analysis tools gave no flexibility to the user, were not easily extendable, and had no unit tests. This module fixes all of that.

3.5.1 Module description

Transition interface sampling (TIS) is a powerful rare events method with a particular focus on calculating the rates of reactions. The core idea starts by splitting the rate k_{AB} into a product:

$$k_{AB} = \phi_{A_0} P_A(B|\lambda_0) \quad (1)$$

where k_{AB} is the rate from state A to state B , ϕ_{A_0} is the flux out of state A and through an interface λ_0 , and $P_A(B|\lambda_0)$ is the transition probability of that a trajectory enters B before any other state given that it has exited the interface λ_0 , starting in state A .

TIS further splits the transition probability into several conditional probabilities, by adding a set of m interfaces (surfaces in phase space) $\{\lambda_i\}$, with λ_0 as the innermost. Mathematically, this gives us:

$$P_A(B|\lambda_0) = P_A(B|\lambda_m) \prod_{i=0}^{m-1} P_A(\lambda_{i+1}|\lambda_i) \quad (2)$$

By sampling trajectories that necessarily cross each given interface λ_i , TIS provides the information that can be used to determine $P_A(\lambda_{i+1}|\lambda_i)$. However, several approaches have been developed/proposed to efficiently turn the sampling data into a best estimate of the transition probability.

The previous analysis in OPS took one of those methods, and provided very little room to customize the procedure. This module makes it so that it is easier to customize the analysis or to use different approaches to calculate the various terms that make up the TIS rate expression.

A much more detailed description of the TIS analysis as implemented here is given in the core OPS documentation, which was also contributed as part of this module. That section of the documentation is online at [this link](#).

3.5.2 Motivation and exploitation

Prior to this module, the TIS analysis in OPS was not very modular, and not tested. This module rewrites the analysis in a modular way, so that the code can be reused and so that the details of the how the analysis is performed can be modified. It also added thorough automated software tests for the analysis process.

Practical exploitation:

- *OPS introduction*: This module will be used in the analysis of TIS simulations for the OPS introductory paper, currently in preparation [21].

Additional information:

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

3.6 Interface optimization

The module provides an iterative approach to optimizing interface locations in TIS.

3.6.1 Module description

Transition interface sampling (TIS) requires sampling trajectory from multiple ensembles. These ensembles must be sampled until to convergence, which means that one would like to have as few ensembles as possible. However, the TIS rate calculation is based on the overlap between successive ensembles, and therefore the ensembles must be "close" enough in trajectory space to have significant overlap. This requirement leads to needing more ensembles in order to cover the same domain.

This trade-off between needed enough ensembles to give sufficient overlap during an entire transition, while desiring as few ensembles as possible for speed, leads to an opportunity for optimization. This module implements such optimization schemes. The procedure for optimization is to first run a short TIS simulation with a initial guess for the interface locations. The results from this short simulation are used to optimize the interface locations by trying to ensure that all interfaces have the same successive crossing probabilities. A new TIS simulation can be run with the new interface locations, and this procedure can be iterated until convergence.

Details on this method as applied to TIS can be found in a paper by Borrero, Weinwurm, and Dellago [25], and its predecessor, applying the approach to FFS, can be found in a paper by Borrero and Escobedo [26].

3.6.2 Motivation and exploitation

A module for interface optimization was identified as a target for E-CAM development in the State of the Art Workshop report. This module was identified as a feasible module for development at the ESDW. Development was driven by the community need and potential for this module to be a training tool at an ESDW, not by a specific scientific application.

Additional information:

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

3.7 Maximum likelihood for reaction coordinates

This module implements an OpenPathSampling library that provides a maximum likelihood analysis (a machine learning approach) to obtain an optimal reaction coordinate by combining multiple collective variables.

3.7.1 Module description

Path sampling methods generate many transition trajectories. However, such trajectories do not automatically lead to a physical understanding of the reaction mechanism. To gain such an understanding it is desirable to find a set of collective variables (CVs) that carry physically important information about the process.

The size and the shape of a crystalline cluster in a freezing liquid, the number of native contacts in a folding protein, or bond length and bond angles in chemical reactions are examples of such CVs. The aim of this module is to find an optimized combination of multiple CVs into a single coordinate, that monitors the progress of the reaction. Such a coordinate is commonly called a reaction coordinate (see also Sec. 3.1).

In methods used to study complex processes, having a good reaction coordinate either significantly improves their efficiency or it is a prerequisite for the reliability of their results.

The reaction coordinate is constructed by optimizing the likelihood function

$$L = \prod_{\text{yes}} p(r(q_i)) \prod_{\text{no}} (1 - p(r(q_i))), \quad (3)$$

where r is a reaction coordinate model that combines several user-proposed CVs, q_i , and p is the probability model that maps this coordinate to a probability of having a successful outcome (yes). The definition of a successful outcome depends on the chosen probability model. Both r and p depend on a set of coefficients that are used to maximize L .

Further details on this method can be found in several papers by Peters and co-workers [27, 28, 29].

3.7.2 Motivation and exploitation

One of the areas that the State of the Art Workshop report identified as a target for E-CAM development was reaction coordinate analysis modules, and the maximum likelihood analysis is one of the better-known ways to extract a reaction coordinate from path sampling data. In addition, this module was identified as a feasible target for development at an ESDW. Development was driven by the need within the community and by the feasibility to use it as part of training at an ESDW, not by a specific scientific project. As a result, it has not yet had any direct practical exploitations.

Additional information:

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

3.8 Channel analysis

In many cases, more than one channel between the states characterizing “reactants” and “products” of a rare event is available to a system — either because there are multiple channels between two states, or because there are multiple states, and transitions between each pair is a different channel. This module provides tools to identify which paths are in each channel, and to study the statistical behavior of the switching between channels.

3.8.1 Module description

This module uses the `OPS Ensemble.split` function to study how a simulation samples multiple channels. The user must provide a list of possible channels, described as `OPS Ensemble` objects. From this, each path is analyzed, and various statistical behavior about the sampling process can be determined.

Once the analysis has been performed, several properties can be extracted, including:

- `switching_matrix`: how many times a switch from one channel to another occurred
- `residence_times`: the number of MC steps spent with the path in each channel (returns the entire list so the user can calculate distribution properties with, e.g., `numpy`)
- `total_time`: total number of MC steps spent in each channel
- `status(step_num)`: the channel the simulation was in for a given step number

3.8.2 Motivation and exploitation

This module was motivated by the needs of several scientific projects, where it was used to understand the sampling efficiency of different channels or different reactions in multiple state systems. In addition, it is an example of an analysis tool for path sampling, which was one of the targets identified by the State of the Art Workshop report.

Practical exploitation:

- *WC-HG DNA*: This module was initially developed to analyze the switching between the “inside” and “outside” mechanisms in the Watson-Crick to Hoogsteen conformational transition in DNA. Due to difficulties translating the data into the OPS format, it was not used for the publication results.
- *KRas dynamics*: This module was used to study the switching between different transitions during the path sampling of wild type and mutant KRas. One of the results was that there was significantly more switching between path types in the wild type, as compared to the mutant. This result will be part of the forthcoming publication from this project [20].

Additional information:

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

3.9 Resampling statistics

The [pandas framework](#) [30] is a powerful and widely-used Python package for analysis of data that can be represented as a 1D series, a 2D matrix (DataFrame), or a 3D “panel.” Many results from OpenPathSampling can be returned to pandas formats, which facilitates integration with other tools, facilitating plotting, visualization, export to \LaTeX tables, and writing to file formats such as Excel or CSV.

This module provides tools for resampling (e.g., statistical bootstrapping) in the general context of functions that return pandas DataFrames, and specifically for OpenPathSampling. This provides tools for estimating the statistical error on multiple quantities simultaneously, such as the rates in the rate matrix of multiple state systems.

3.9.1 Module description

Providing an estimate of uncertainty is essential when presenting scientific results. This module provides the ability to perform statistical analysis of a large simulation from OpenPathSampling by using the sampled trajectories to create subsamples, which are then assumed to be independent. The subsamples are analyzed separately, and this module makes it easy to obtain mean, standard deviation, or percentile values. In particular, this module provides the tools to do such an analysis on functions that return a table of data using a pandas.DataFrame object, as the OpenPathSampling rate matrix calculation does.

Most of the code is generic, and could be used for any function that produces a pandas.DataFrame as its output. Therefore this module may be useful for many projects other than OpenPathSampling. Within OpenPathSampling, this can be used to obtain statistics on rates, fluxes, and other such quantities.

While BlockResampling is the only resampling method implemented in the module (as it is the one needed for TIS rate calculations), it would be straightforward to extend this framework with other resampling methods, such as variants of bootstrapping.

3.9.2 Motivation and exploitation

Many of results from OpenPathSampling are returned using pandas.DataFrame objects to hold matrices. This module provided the necessary ability to calculate uncertainties based on that. It was developed based on the need to calculate these uncertainties as part of the paper introducing OpenPathSampling [21].

Practical exploitation:

- *OPS introduction*: This module is used to calculate error estimates in the examples from the introductory paper on OPS, currently in preparation [21].

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 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. Many of the modules in this report focus on analysis, following the description of this Deliverable in the grant agreement, and therefore scalability does not play a major role. But some of the modules in this report include both sampling approaches and analysis, and therefore they can benefit from improved sampling performance.

In E-CAM Deliverable D1.1, 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 the committor analysis, the reactive flux, and the S-shooting algorithm wrap around such dynamics engines, and their performance characteristics therefore depend on the underlying engine. The objects created by 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. In addition, the transition state ensemble and interface optimization modules, which wrap around the entire simulation process, also inherit the performance of the underlying engine.

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.

In addition, we remark that parallelization by running multiple simultaneous trajectories is in development as future E-CAM modules. Because of the inheritance model used in developing these modules, adding the ability to run multiple trajectories to, e.g., the committor analysis, will automatically result in the that ability being present in reactive flux, S-shooting, and the transition state ensemble calculation.

5 Outlook

The report of Deliverable 1.3 of E-CAM describes 9 software modules in classical dynamics. They include codes and documentation for tools for machine learning and statistical analysis of rare events. These modules interface with, and many are included in, the open-source package OpenPathSampling. Eight of these modules have been fully accepted into the E-CAM software library, and the remaining module will be completed soon.

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.

Most of these modules relate to the statistical analysis of rare events, although the maximum likelihood prediction of reaction coordinates is a machine learning method, and the committor analysis is a prerequisite for performing the machine learning approach of Ma and Dinner [22].

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, and to understand the difference between the dynamics of wild-type and oncogenic forms of KRas. In addition, several of these modules provide basic examples of common path sampling tools in the OPS package. Several of these modules were explicitly listed as targets for E-CAM development in either Deliverable D1.1 or the State of the Art Workshop report, including committor analysis, reactive flux, transition state ensemble, and interface optimization. Many other modules fall into other categories for development cited in the State of the Art Workshop report⁶, in particular the categories of “reaction coordinate analysis modules” (maximum likelihood for reaction coordinates) and “analysis tools to work with path sampling” (TIS analysis framework, channel analysis, and resampling statistics). The modules delivered here respond to real needs of the classical MD community, and are of practical use for exploitation by academia and industry.

Several modules from this report demonstrate the success of the ESDWs that have been organized by E-CAM WP1. The modules for reactive flux, transition state ensemble, S-shooting, maximum likelihood for reaction coordinates, and interface optimization were all developed in the context of the E-CAM ESDWs, with most coming from the 2016 ESDW in Traunkirchen, Austria, and S-shooting coming from the 2017 ESDW in Leiden, Netherlands. The ESDW in Traunkirchen also included two modules (shifting move and aimless shooting) that did not fit the focus of this report, and therefore will be part of a later deliverable. Additional modules from the August 2017 ESDW in Leiden are still in development, and will be part of later deliverables.

Future work will extend several of these modules by simultaneously running multiple trajectories. This will, in a single module, provide significant scalability for several of the modules presented here, including the committor analysis, reactive flux, S-shooting, and transition state ensemble. These modules can already make use of parallelization within a single trajectory. They were developed without the more advanced level of parallelization in order to thoroughly test them in the simpler environment. Modules for this more advanced parallelization will be provided in future deliverables.

⁶Report 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

CV collective variable

ESDW Extended Software Development Workshop

HG Hoogsteen

MD molecular dynamics

OPS OpenPathSampling

TPS transition path sampling

TIS transition interface sampling

WC Watson-Crick

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/131>
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>
<https://www.cecama.org/workshop-0-1356.html> ... <https://www.cecama.org/workshop-0-1356.html>
<https://www.cecama.org/workshop-0-1406.html> ... <https://www.cecama.org/workshop-0-1406.html>
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/>

Page 3

OpenPathSampling GitHub repository ... <http://github.com/openpathsampling/openpathsampling>
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/aaronkeys/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 6

<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_committor_analysis/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/6

Page 7

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_reactive_flux/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/16
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/45

Page 8

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_tse/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/9

Page 9

this link ... http://openpathsampling.org/latest/topics/tis_analysis.html
Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_new_tis_analysis/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/38
Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_interface_optimization/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/17

Page 10

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_maxlikelihood/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/29

Page 11

Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_channel_analysis/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/25
pandas framework ... <https://pandas.pydata.org/>
Module documentation ... http://e-cam-classical-md-modules.readthedocs.io/en/latest/modules/OpenPathSampling/ops_resampling_statistics/readme.html
Documentation merge request ... https://gitlab.e-cam2020.eu/E-CAM/Classical-MD-Modules/merge_requests/37

Page 13

<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] 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.

- [22] Ao Ma and Aaron R. Dinner. Automatic Method for Identifying Reaction Coordinates in Complex Systems. *J. Phys. Chem. B*, 109:6769, 2005.
- [23] Titus S van Erp and Peter G Bolhuis. Elaborating transition interface sampling methods. *J. Comput. Phys.*, 205(1):157, 2005.
- [24] Georg Menzl, Andreas Singraber, and Christoph Dellago. S-shooting: a Bennett-Chandler-like method for the computation of rate constants from committor trajectories. *Faraday Discuss.*, 195:345, 2016.
- [25] Ernesto E Borrero, Marcus Weinwurm, and Christoph Dellago. Optimizing transition interface sampling simulations. *J. Chem. Phys.*, 134(24):244118, 2011.
- [26] Ernesto E Borrero and Fernando A Escobedo. Optimizing the sampling and staging for simulations of rare events via forward flux sampling schemes. *J. Chem. Phys.*, 129(2), 2008.
- [27] Baron Peters and Bernhardt L Trout. Obtaining reaction coordinates by likelihood maximization. *J. Chem. Phys.*, 125(5):054108, 2006.
- [28] Baron Peters, Gregg T Beckham, and Bernhardt L Trout. Extensions to the likelihood maximization approach for finding reaction coordinates. *J. Chem. Phys.*, 127(3):034109, 2007.
- [29] Baron Peters. Reaction Coordinates and Mechanistic Hypothesis Tests. *Annu. Rev. Phys. Chem.*, 67:669, 2016.
- [30] pandas: Python Data Analysis Library, 2017. URL <https://pandas.pydata.org/>. [Online; accessed 20-November-2017].