Classical MD E-CAM Modules IV

E-CAM Deliverable 1.5
Deliverable Type: Report
Delivered in February 2020

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	D1.5
Deliverable Nature	Report
Dissemination Level	Public
Contractual Date of Delivery	Project Month 48(30th September, 2019)
Actual Date of Delivery	28/02/2020
Description of Deliverable	9 software modules delivered to the E-CAM repository in the area of Classical Molecular Dynamics responding to requests of users, and their documentation.

Document Control Information

Document	Title:	Classical MD E-CAM Modules IV
	ID:	D1.5
	Version:	As of 28th February, 2020
	Status:	Accepted
	Available at:	https://www.e-cam2020.eu/deliverables
	Document history:	Internal Project Management Link
Review	Review Status:	Reviewed
Authorship	Written by:	David W.H. Swenson (École Normale Supérieure de Lyon)
	Contributors:	Pascal Carrivain (École Normale Supérieure de Lyon), Sarath Menon (Ruhr University Bochum), Andreas Singraber (University of Vienna)
	Reviewed by:	Ana Mendonça (EPFL), Alan O'Cais (JSC)
	Approved by:	Ana Mendonça (EPFL), Alan O'Cais (JSC)

Document Keywords

Keywords:	E-CAM, Molecular Dynamics, CECAM, Rare Events, Path Sampling, OpenPathSampling

28th February, 2020

***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.*

[1] dwhs@hyperblazer.net

Contents

Executive Summary

In this report for Deliverable 1.5 of E-CAM, 9 software modules in classical dynamics are presented. These modules represent improvements and new features in response to the needs of users.

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

- To implement the recommendations contained in E-CAM Deliverable D1.1 (Identification/selection of E-CAM MD codes for development) [1] and by the reports of E-CAM Classical MD State of the Art Workshops, the first held 29 August – 2 September 2016 at the Lorentz Center in Leiden, Netherlands, and the second held 1–3 October 2018 at the Erwin Schrödinger Institute in Vienna, Austria.[2]
- 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 user requests and to the scientific needs of several active research projects, such as the construction of coarse-grained models based on neural network potentials, and assisting users in the simulation and analysis of polymer models.

The modules in this report include several based on OpenMM, OpenPathSampling (OPS) and n2p2. OpenMM is a powerful GPU-accelerated library, and the OpenMM-based modules gain their performance by using it. OpenPathSampling wraps around other dynamics engines, such as OpenMM, and leverages the performance that is already developed in those. n2p2 is a software package supporting parallelization via Message Passing Interface (MPI) and Open Multi-Processing (OpenMP). In addition, a Python interface to n2p2's underlying C++ libraries is provided. The presented modules based on n2p2 inherit and enhance the performance of the package. Another module presented here uses Python bindings for C++ code in order to provide the flexibility of Python while retaining the performance of C++.

The 9 modules presented here are:

1. Gromacs engine for OPS
2. VisitAllStatesEnsemble for OPS
3. Interface-Constrained Shooting in OPS
4. Double-Well Test Systems in OpenMMTools
5. OpenMM Plectoneme
6. OpenMM Copolymer
7. pyscal
8. NNP-CG - Descriptor analysis
9. n2p2 - Symmetry Function Memory Footprint Reduction

Each module is thoroughly tested, includes in-code documentation as well as external documentation, frequently in the form of Jupyter notebook examples.

Section 1 of this report gives a brief description of E-CAM modules and the role of this deliverable in the broader goals of E-CAM Work Package 1 (WP1). Section 2 provides background material on the contexts shared between multiple modules: rare events and applications path sampling, simulation setup tools, and neural network potentials. In section 3, we describe each of the modules and provide links to their documentation. Section 4 describes performance aspects of these modules, and section 5 summarizes the deliverable and describes the outlook for future development of modules within WP1, including the increasing importance of transverse actions across simulation communities and work packages.

[2] State of the Art Workshop reports available for download from the E-CAM website: https://www.e-cam2020.eu/scientific-reports/

1 Introduction

Notwithstanding the exponential increase in computing power over the last few decades and the development of efficient molecular dynamics algorithms, many processes are still beyond the reach of simulation, especially those associated with very long and disparate timescales. For instance, the folding of a protein may occur on the time scale of seconds, or a liquid can exist in an under-cooled state almost indefinitely, but the fastest motion may correspond to a time of the order of the femtosecond. Moreover, when the folding or the freezing occurs, it does so quickly. That is, events of interest are not slow but rather rare, causing long waiting times before they can be observed, and require an impractical number of time steps to be simulated directly. A similar problem occurs when interaction models beyond conventional force fields are desired. In principle *ab initio* methods offer the possibility to describe systems which are unsuited for empirical potentials, but their use in large-scale molecular dynamics simulations is severely hampered by the high computational cost and the unfavorable scaling behavior.

Addressing such time and size scale problems and developing scientific software able to overcome them is one of the central goals of Work Package 1 (WP1) of the E-CAM-Project. It does so by providing academic and industrial users the means to address such questions using open source software with verified quality standards, appropriate documentation and testing, disseminated and in part generated through state of the art workshops; industry scoping workshops; Extended Software Development Workshop (ESDW) events; and industry pilot projects. E-CAM software is produced primarily through E-CAM pilot industry projects funded directly by E-CAM, and through the ESDW's. ESDW's typically of 1–2 weeks duration are a unique approach to combine software development with training, i.e. "training by doing", and of engaging with the wider simulation community.

In E-CAM Deliverable D1.1 [1], we gave an overview of existing software for rare events and future needs, and highlighted areas where E-CAM could make useful contributions. That included the development of modules for rare events methods, particularly in the context of the OPS package. More recently, we have revisited the needs of the community through the State of the Art Workshops. In addition to a continued interest in path sampling, those workshops emphasized the growing importance of artificial neural networks. The use of neural networks to approximate *ab initio* force calculations has been repeatedly highlighted. These considerations, as well as the explicitly expressed interests of ESDW participants and the needs arising from new collaborations with experimental researchers, have guided the selection of the modules listed here. In this way, these modules are responding to the requests of users.

1.1 This Deliverable in the context of E-CAM

This report covers the fourth group of nine modules delivered as part of E-CAM WP1. As described in the grant agreement, they are "in the area of classical molecular dynamics responding to requests of users, and their documentation."

Seven of these modules were produced by E-CAM Postdoctoral Research Associate (PDRA)s and two were produced by participants at E-CAM ESDWs, one from the 2017–2018 ESDW in Leiden, Netherlands, and one from the 2018–2019 ESDW in Turin, Italy. Additional modules have also been developed which are not part of this deliverable.

As mentioned above, this report includes two modules that were contributed from ESDWs. Selection of modules for the ESDW was driven by the combination of feasibility and relevance to the goals of the Deliverable and of the project, as laid out in previous reports. These software development tasks were also 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, as well as their intrinsic software value.

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.

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 path sampling modules in this report are: a module to interface with other software used by the E-CAM community (Gromacs engine for OPS), a module implementing an efficient path sampling algorithm (Interface-Constrained Shooting in OPS), and a module to improve usability during the setup phase of an OPS simulation (VisitAllStatesEnsemble for OPS).

2.2 Simplifying simulation setup

Several of the modules reported here are designed to facilitate simulation setup. These modules represent a wide range of scientific applications, but are linked in that they make it easier to set up some kind of simulation, using OpenMM for the molecular dynamics.

Classical dynamics simulations of non-atomistic models can play an important part in molecular simulation, including in multiscale modelling. Additionally, simple models are frequently used to develop new methods. By implementing the models using OpenMM, these modules gain the GPU performance and integration with other tools that comes from OpenMM.

The modules that assist with simulation setup in OpenMM are Double-Well Test Systems in OpenMM, OpenMM Plectoneme, and OpenMM Copolymer.

Additionally, OpenMM Plectoneme, and OpenMM Copolymer assist with simulation analysis and some of the analysis use Dask and Numba to speed-up the computation.

In the context of simplified simulation setups the module "NNP-CG - Descriptor analysis" should also be mentioned. The tools in that module support the search for a well-balanced set of atomic environment descriptors. This setup step is essential in the process of creating new high-dimensional neural network potentials because the set composition can be a limiting factor for the overall predictive power.

2.3 Neural network potentials

Many systems in computational physics and chemistry can be successfully studied with empirical force fields at the atomistic level. In the context of these "molecular mechanics" models, atoms are treated as particles without internal structure and their interactions are defined via rather simple expressions deduced from physical/chemical intuition. Usually a small number of free parameters is enough to tune the potential to reproduce experimental properties with good agreement. However, there are systems for which a satisfying description within this framework is not possible. Take as an example the formation and breaking of covalent bonds. This is the territory of *ab initio* methods which use quantum mechanics to accurately model the behavior of the system. Unfortunately the additional level of detail

comes at a cost. Even in small systems *ab initio* methods are usually many orders of magnitude slower than empirical force fields. Moreover, the computational cost increases unfavorably with the number of atoms which makes it impractical to perform large simulations.

With rising influence of machine learning algorithms in science and technology a new category of interatomic potentials has emerged. Machine learning potentials (MLPs) aim at bridging the gap between *ab initio* methods and empirical force fields. In contrast to the latter, MLPs are not bound by a predetermined fixed functional form of the interaction but rather build on the flexibility of an underlying machine learning model, such as artificial neural networks. These are known for their capability to reproduce any complicated function, which in this case is the desired potential energy surface, but rely on a separate training stage before they are ready for use. During this phase the MLP "learns" from a large data set how energies and forces depend on atomic positions. The reference energy landscape is typically computed from expensive *ab initio* methods. Once the training is completed the MLP can accurately predict energies and forces for new (unseen during training) atomic configurations at a fraction of the cost of the reference method. Hence, with MLPs times scales become accessible in molecular dynamics simulations close to those of empirical potentials while maintaining the *ab initio* level of accuracy.

Today MLPs exist in various forms and combine different atomic environment descriptors as inputs for all kinds of machine learning models. A very successful variant is the high-dimensional neural network potential (HDNNP)[3]. The software n2p2 implements the method, provides tools for training and supplies an interface to the popular molecular dynamics package LAMMPS. Two n2p2 related modules are reported below: one presents additional analysis tools, and a second one tackles the problem of high memory usage.

3 Modules

The modules described here are based on several software packages. The first three modules listed are are based on OPS, adding substantial new user-requested functionality and improving usability. The next three modules are based on OpenMM. They facilitate the setup of new simulations, and are designed to benefit from the underlying performance, especially on GPUs, of OpenMM. The seventh module, pyscal, was a contribution from the 2018–2019 ESDW in Turin, which created Python bindings around C++ code to give users the flexibility of Python with the performance of C++. Finally, there are two modules which extend the capabilities and performance of n2p2. The first one builds upon existing code infrastructure and provides additional analysis tools, whereas the other one drastically reduces the memory usage via modification of the core library.

Material in this section is largely drawn from the detailed module documentation files hosted at Classical MD section of the E-CAM Library. Links are provided for more information about each module. Modules that have been completed and accepted into the E-CAM library 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 Gromacs engine in OPS

This module adds support for Gromacs as an engine for OpenPathSampling.

3.1.1 Module description

Different molecular dynamics (MD) codes have developed to serve different communities. Gromacs is one of the major MD codes for the biomolecular community, and even though much of its functionality can be reproduced by other MD codes, such as OpenMM, there are still some extensions that are built on top of Gromacs that haven't been ported to other codes. For example, the MARTINI coarse-grained model is not available on other codes such as OpenMM.

Additionally, people who are familiar with a given MD package will prefer to continue to work with that. Therefore codes that wrap around MD packages, as OpenPathSampling does, can expand their reach by adding ways to interface with other MD packages.

This module adds the Gromacs engine for OpenPathSampling. It is the first practical test of the external engine API of OPS.

3.1.2 Motivation and exploitation

Gromacs support has been one of the most-requested features in OpenPathSampling. This makes OPS available to a new audience of users. Gromacs has much better performance on CPUs than OpenMM, which OPS already supported. In addition, there are some tools (such as the MARTINI coarse grained force field) that are available in Gromacs, but not in OpenMM.

This module has been exploited by at least two projects:

- A bachelor's thesis project at the University of Amsterdam, studying the opening and closing of T4 lysozyme
- A project involving membrane proteins simulated using the MARTINI forcefield, by researchers at the Max Planck Institute of Biophysics

Additional Details

Direct Documentation Link	readme.rst of Gromacs engine in OPS module.
Merge Request	Merge Request of Gromacs engine in OPS module.

3.2 OPS Visit All States Ensemble

This module adds a convenient new OpenPathSampling ensemble that allows trajectories to continue until they have visited all the states in the system. In addition, it provides real-time reporting about the progress.

3.2.1 Module description

One of the ways to get initial trajectories for path sampling is to use dynamics that aren't physical for the ensemble of interest, such as using an increased temperature. If a trajectory has a frame in every state, then it must have subtrajectories that connect from each state to another one, and therefore it has all the information to start a MSTIS simulation. The ensemble definition tools in OPS make it easy to create such custom sequential ensembles [4].

However, users often want to do simulation setup in an interactive mode, such as in a Jupyter notebook, or at a minimum want to have a sense of the progress made on a long trajectory such as this. The default OPS ensemble gives no output and therefore no sense of how much progress has been made.

This module provides a custom OPS ensemble that gives such output during its simulation. It outputs the length of the trajectory so far, as well as the states that have and have not already been visited. This gives a much better sense of how long the simulation will take to run.

3.2.2 Motivation and exploitation

The essential functionality of this had been in one of the OpenPathSampling example notebooks, but it was shown in a way that was neither simple nor reusable. Creating a module that simplifies this task, and which is not dependent on the specific system being studied, both makes the example easier to understand and makes the tool reusable for others.

This module is now used in the OPS alanine dipeptide MSTIS example.

Additional Details

Direct Documentation Link	readme.rst of OPS Visit All States Ensemble module.
Merge Request	Merge Request of OPS Visit All States Ensemble.

3.3 Interface-Constrained Shooting in OPS

This module adds interface-constrained shooting to OpenPathSampling. Interface-constrained shooting is a technique that can improve the efficiency of transition interface sampling.

3.3.1 Module description

In transition interface sampling (TIS), one defines stable states (volumes in phase space) and interfaces (surfaces in phase space). For a trajectory to be accepted in TIS, it must begin with exactly one frame in a given initial state, cross the interface, and end with exactly one frame in any state volume (including the initial state).

New trajectories are generated with the shooting move, which selects a point along an initial trajectory from which new frames can be made. In one-way shooting, the dynamics only needs to run in one direction (with the stochastic nature of the dynamics ensuring that a new trajectory is generated).

However, if the interfaces are far from the initial state and if all frames are equally likely to be used for shooting, it can be very likely for the shooting point to come before the trajectory has crossed the interface. This can then lead to shooting moves that usually generate trajectories that don't cross the interface, and therefore must be rejected. This uses a lot of simulation effort without generating useful new trajectories.

Interface-constrained shooting (also called "constrained interface shooting", see [5]) is an approach to solve this problem. Instead of selecting from anywhere along the trajectory, only the first point after crossing the interface is allowed as a shooting point. This ensures that every trajectory that is generated will be valid (will cross the interface). In addition, because the first crossing is still the first crossing in the new trajectory, this leads to the Metropolis acceptance probability also being 1. Therefore, every trial trajectory is accepted.

In practice, this must be combined with the path reversal move in order to sample all of trajectory space. The result is an approach with very high acceptance, although decorrelation of the trajectory is a little slower.

3.3.2 Motivation and exploitation

This is a tool that can make transition interface sampling simulation much more efficient, in certain circumstances. It was implemented as part of an ESDW.

Additional Details

Direct Documentation Link	readme.rst of Interface-Constrained Shooting in OPS module.
Merge Request	Merge Request of Interface-Constrained Shooting in OPS module.

3.4 Double-Well Dimer Testsystems

One of the common systems used to study rare events is the double-well dimer in a bath of repulsive particles. In this system, two particles are linked by a "bond" that allows condensed and extended metastable states. This module adds this system, and tools for created several variants of it, to the OpenMMTools package.

3.4.1 Module description

The symmetric double-well dimer is a widely-used model for developing new rare events methodologies. However, implementing simple models in software packages that are designed for biological systems, such as OpenMM, can be difficult for a novice user. As a result, many developers of new methods will implement their methods twice: first to interface with simple models such as the double-well dimer using in-house MD codes, then a second time to interface with more powerful tools, such as OpenMM, to simulate complex systems such as biomolecules. This module provides tools that facilitate setting up custom versions of the double-well dimer for OpenMM, allowing users to develop their new methodologies directly for the same platform that they will use for larger practical applications.

The widely-used double-well dimer model is a symmetric quartic potential, given by:

$$V_{dw}(r) = h\left(1 - \left(\frac{r - r_0 - w}{w}\right)^2\right)^2$$

where r is the distance between the particles, h is the height of the barrier, r_0 is the energy minimum for the condensed metastable state, and w sets the distance for the extended metastable state according to $r_{ex} = r_0 + 2w$.

This "bonded" interaction is added for specific pairs of particles, on top of a background of WCA (purely repulsive) "nonbonded" interactions between all particles. The WCA interaction is:

$$V_{\mathrm{WCA}}(r) = \begin{cases} 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right) + \epsilon & \text{if } r \le 2^{1/6}\sigma \\ 0 & \text{if } r > 2^{1/6}\sigma \end{cases}$$

where σ is a characteristic distance and ϵ is a characteristic energy scale.

The quartic double well is a simple model of rare events, where the expected reaction coordinate is obvious. However, it is can be very useful for benchmarking new methods. The OpenMMTools package includes a suite of systems to be used in testing and benchmarking, and was a natural place to add these.

Although the most widely-used approach has been to have a single dimer in the bath of WCA particles, this module provides two possible extensions that have been previously used in the literature. The first extension is to allow multiple independent dimers, as was done in [6]. This is done by changing the `ndimers` parameter in the `DoubleWellDimer_WCAFluid` test system. The other extension is to create a polymer chain of double-well bonds, as was done in [7]. This is done by changing the `nchained` parameter in the `DoubleWellChain_WCAFLuid` test system.

3.4.2 Motivation and exploitation

This module is intended for methods developers, allowing a single code base to be used for both toy model testing and real production simulations.

Additional Details

Direct Documentation Link	readme.rst of Double-Well Dimer Testsystems module.
Merge Request	Merge Request of Double-Well Dimer Testsystems module.

3.5 OpenMM Copolymer

OpenMM_Copolymer is a module that samples conformations of a block-copolymer given a genome epigenetic state file. This module takes advantage of the OpenMM software and GPU acceleration. It builds a Kremer-Grest polymer model with uni-dimensionnal epigenetic information and constructs the epigenetic interactions based on the model you design. You simply need to feed the module with a epigenetic state file, the interaction model, and the mechanical properties of the polymer.

3.5.1 Module description

Recently, the epigenetic and the tri-dimensional structure of fly genome has been studied by means of *block-copolymers*. The *block-copolymer* is made of more than one monomer species. The epigenetic information does not involve alterations in the DNA, but histone tail modifications. This uni-dimensional information can be projected along the contour of a *block-copolymer* model. Then, every pair of monomers interacts according to the epigenetic states leading to specific pattern of interactions. The interaction patterns can be visualized using a contact map : two-dimensional map with position along the polymer and a third dimension with color scale for the intensity of contacts. Since 2000, biologists have been producing the same kind of data thanks to the *high-throughput-sequencing* methods 3C, 4C, 5C and Hi-C : Chromosome-Conformation-Capture. Recently, biologists have shown that the interaction pattern is correlated with the epigenetic information. However, the strength and model of interactions between epigenetic states are not always clearly known.

The module we propose uses the OpenMM software with GPU acceleration to sample as many epigenetic parameters as possible. It is possible to use effective interactions (Gaussian overlap or Lennard-Jones potential) to model the epigenetic interactions. This module introduces the possibility to replace effective epigenetic interactions with a binders model too. In this case, the binder is like a protein that can bind to a specific site of the genome. A simple input file is enough to tell the script about the binder-binder and monomer-binder interactions.

The present module assists with simulation of a *block-copolymer* model and assists with the analysis of the data (HPC Dask and Python compiler Numba). It can be used by polymer physicists and biophysicists for epigenetic modeling, to understand the link between epigenetic and tri-dimensional structure of a genome, to estimate first-passage-time encounter of two loci. It is being used in a scientific collaboration to study a specific promoter-enhancer system in the fruit-fly organism (Yad Ghavi-Helm and Cedric Vaillant, ENS Lyon, France).

3.5.2 Motivation and exploitation

The idea of this module starts from a collaboration with a group of biologists from ENS Lyon. They wanted to study by means of numerical experiments the interaction between specific promoter and enhancer and confronts it with experiments. However, such a system has to be studied at very small scale (base-pair resolution). The OpenMM API is used to assist the creation and because of computational efficiency. Eventually, the module can be used by any physicists with interest in the *block-copolymer* simulation for epigenetic modelling.

Additional Details

Direct Documentation Link	OpenMM Copolymer module
Merge Request	Merge Request of OpenMM Copolymer module.

3.6 OpenMM Plectoneme

The OpenMM_Plectoneme is a module that introduce twist to a ring or linear polymer and sample the accessible conformations under torsional constraints. This module takes advantage of the OpenMM software and GPU acceleration to perform simulations at the scale of the DNA helix. It builds a Kremer-Grest polymer model with virtual sites to attach a frame to each of the bead. The frames are used to describe the contour of the molecule and to introduce bending and twisting forces.

3.6.1 Module description

Bacterial DNA is known to form specific conformations called *plectonemes* because of internal twisting constraints. This physical mechanism participates in the compaction of the genome. In order to study such a system we need

to introduce a linking number deficit into a circular polymer. The Linking number (Lk) is the sum of the Twist (Tw, cumulative helicity of the DNA) and the Writhe (Wr, global intracity). In the case of circular DNA that is topologically constrained, any variation of the Twist affects the Writhe and therefore the conformation. In particular, does a slow change of the twist lead to the same conformation that we get from a rapidly changing Twist? We then tackle the question: does the introduction protocol of Linking number inside circular molecule matter? Indeed, does a rapid Linking number injection freeze the conformation in braided structures where *plectonemes* do not merge/move along the DNA ? Does the memory of initial conformation matter ?

We can use this module to model single-molecule DNA under magnetic or optical tweezers too. In this setup the molecule is clamped on a plate and to a magnetic bead at the other extremity. The bead is used to apply a stretching force and/or rotational constraint. The position of the bead is used to monitor the response of the molecule to the mechanical constraints. From the mechanical constraints you can extract the mechanical properties of your molecule of interest.

This module assists the creation of a polymer described by FENE bond and WCA repulsive potential to resolve the excluded volume constraints. On top of that, the module introduces the twist and mechanical response to twisting constraint with the help of *virtual sites* functionality from OpenMM API. It provides functions for data analysis using numba for performance and dask for high-throughput computing. For example, the estimation of the Writhe that is a computation over all the possible pairwise of bonds is highly expensive and can be fasten. In addition to that, we introduce an algorithm to detect the positions, length and shape of *plectonemes*. It is useful to follow the dynamics of these braided structures and try to answer the previous questions.

This module can be used by polymer physicists to understand the conformation of bacterial DNA under torsional constraints, for example. It is currently used in a scientific collaboration with Ivan Junier from TIMC-IMAG, Grenoble, France and Ralf Everaers, ENS Lyon, France.

3.6.2 Motivation and exploitation

The idea of this module starts from a discussion with a biophysicist from TIMC-IMAG, Grenoble, France. Indeed, the home-made code he developed had some computational efficiency issues. This module is intended to ease the setup of twist using the OpenMM API while using the computation power on GPUs hardware to run dynamics of braided structures called *plectonemes*. It implements functions that can be used to help the analysis of the simulation results with HPC Dask and Python compiler Numba.

Additional Details

Direct Documentation Link	OpenMM Plectoneme module
Merge Request	Merge Request of OpenMM Plectoneme module.

3.7 pyscal

`pyscal` is a Python module for the calculation of local atomic structural environments including Steinhardt's bond orientational order parameters [8] during post-processing of atomistic simulation data. The core functionality of pyscal is written in C++ with python wrappers using pybind11 which allows for fast calculations and easy extensions in python.

3.7.1 Module description

Steinhardt's order parameters are widely used for the identification of crystal structures [9]. They are also used to distinguish if an atom is in a solid or liquid environment [10]. pyscal is inspired by the BondOrderAnalysis code, but has since incorporated many additional features and modifications. The pyscal module includes the following functionalities:

- calculation of Steinhardt's order parameters and their averaged version [11].
- links with the Voro++ code, for the calculation of Steinhardt parameters weighted using the face areas of Voronoi polyhedra [9].
- classification of atoms as solid or liquid [10]
- clustering of particles based on a user defined property.

- methods for calculating radial distribution functions, Voronoi volumes of particles, number of vertices and face area of Voronoi polyhedra, and coordination numbers.

3.7.2 Motivation and exploitation

Bond orientational order parameters are a widely-used tool, but challenging to implement. This module, designed and developed by an ESDW participant, provides a powerful and efficient tool to calculate them. The development of this module led to a publication documenting the software [12].

Additional Details

Direct Documentation Link	readme.rst of pyscal module.
Merge Request	Merge Request of pyscal module.

3.8 NNP-CG - Descriptor analysis

This module adds tools to the n2p2 package which allow to assess the quality of atomic environment descriptors. This is particularly useful when designing a neural network potential based coarse-grained model (NNP-CG).

3.8.1 Module description

Creating a coarse-grained (CG) model from the full description of a system is a two-step process: (1) selecting a reduced set of degrees of freedom and (2) defining interactions depending on these coarse-grained variables. For example, in a common coarse-graining approach for molecular systems the atomistic picture is replaced by a simpler description with CG particles sitting at the center-of-mass coordinates of the actual molecules. The corresponding interactions between CG sites can be modelled with empirical force fields but also, as has been recently shown in [13] and [14], with machine learning potentials. This module is the first part of a series to implement coarse-grained models in n2p2 and to provide tools to estimate the quality of atomic environment descriptors, which in turn hints on the expected performance of the coarse-grained description.

The overall goal of the analysis is to show qualitatively whether there is a correlation between the raw atomic environment descriptors (and their derivatives) and the atomic forces. If no or very little correlation can be found we can assume that the descriptors do not encode enough information to construct a (free) energy landscape. On the other hand, if "similar" descriptors correspond to "similar" forces there is a good chance that a machine learning algorithm is capable of detecting this link and a machine learning potential can be fitted. In order to find a possible correlation between descriptors and forces the following approach is used: First, a clustering algorithm (k-means or HDBSCAN) searches for groups in the high-dimensional descriptor space of all atoms. Then, for every detected cluster the statistical distribution of the corresponding atomic forces is compared to the statistics of all remaining atomic forces. A hypothesis test (Welch's t-test) is applied to decide whether the link between descriptors and forces is statistically significant. The percentage of clusters which show a clear link is then an indicator for a good descriptor-force correlation.

In order to perform the analysis described above n2p2 was extended by two separate software pieces:

- **A new application based on the C++ libraries:** `nnp-sfclust`

  This application allows to generate files containing the atomic environment data required for the cluster analysis.

- **A new Jupyter notebook with the actual analysis:** `analyze-descriptors.ipynb`

  The script depends on common Python libraries (`numpy`, `scipy`, `scikit-learn`) and reads in data provided by `nnp-sfclust`. It then clusters the data, performs statistical tests and presents graphical results.

3.8.2 Motivation and exploitation

The software in this module will be used in the context of the WP1 pilot project: Implementation of neural network potentials for coarse-grained models. In addition, the new functionality provides a link to heavily used machine learning software, such as `scikit-learn`.

Additional Details

Direct Documentation Link	readme.rst of NNP-CG - Descriptor analysis module.
Merge Request	Merge Request of NNP-CG - Descriptor analysis module.

3.9 n2p2 - Symmetry Function Memory Footprint Reduction

This module improves memory management in n2p2. More specifically, a new strategy to store symmetry function derivatives is implemented. In this way the memory footprint during training is drastically reduced.

3.9.1 Module description

Training high-dimensional neural network potentials means to minimize the error between predictions and the reference information in a data set of atomic configurations. There, the desired potential energy surface is supplied in the form of an energy per configuration and forces on each atom. Consider the HDNNP expression for forces

$$F_{i,\alpha} = -\sum_{j=0}^{N_{\text{atoms}}} \sum_{k=0}^{N_{\text{sym.func.}}} \frac{\partial E_j}{\partial G_{j,k}} \frac{\partial G_{j,k}}{\partial x_{i,\alpha}}, \tag{1}$$

where $G_{j,k}$ denotes the k-th symmetry function of atom j. Only the first expression $\frac{\partial E_j}{\partial G_{j,k}}$ depends on the neural network weights and therefore changes during the training process. The symmetry function derivatives with respect to atom coordinates $\frac{\partial G_{j,k}}{\partial x_{i,\alpha}}$, however, stay fixed for each atomic configuration in the data set. Given the high computational cost of symmetry functions it is essential to pre-calculate and store them in memory. While this strategy speeds up the training procedure significantly [15] it also drastically increases the memory footprint, which easily reaches more than 100 GB for common data set sizes.

This module alters the core C++ library of n2p2 in order to reduce the memory consumption of all depending applications and provides benchmark results quantifying the improvement. The idea is to exploit the fact that for specific combinations of neighboring atoms i and j the expression $\frac{\partial G_{j,k}}{\partial x_{i,\alpha}}$ always equals zero. Consider a three-component system with elements A, B and C. In addition, let atoms i and j be of element A and B, respectively. Then, the derivative of a symmetry function $G_{j,k}$ with signature B-C (i.e. only sensitive to neighbor atoms of type C) with respect to i's coordinates vanishes. Hence, by taking these element combination relations automatically into account a significant portion of the memory usage can be avoided. Depending on the symmetry function setup, savings of about 30 to 50% can be achieved for typical systems.

Code changes cover most of the classes in the `libnnp` core library where they add functionality to identify relevant (nonzero) element combinations for the symmetry function derivative computation. Additional CI tests ensure that results are not affected.

3.9.2 Motivation and exploitation

This module is based on a user request for reduced memory consumption. The improvements will become particularly relevant in view of future application to systems with three or more elements. The positive effects originating from the changes in the core library propagate to all depending applications and interfaces to third-party software.

Additional Details

Direct Documentation Link	readme.rst of n2p2 - Symmetry Function Memory Footprint Reduction module.
Merge Request	Merge Request of n2p2 - Symmetry Function Memory Footprint Reduction module.

4 Performance Considerations

These modules include a number of different approaches to enhance their performance.

The modules based on OpenPathSampling use its model of leveraging the performance of the underlying molecular dynamics engine. Indeed, one of the modules presented here adds Gromacs as another molecular dynamics engine that OPS can use. This module means that OPS can immediately benefit from the significant performance engineering that the Gromacs developers have done.

Like OpenPathSampling, the modules that build on OpenMM benefit from the performance of the underlying library. OpenMM is highly optimized for running on GPUs, and these modules get that performance "for free" by using the OpenMM library. Similarly, the descriptor analysis tool in the first module for n2p2 gains its performance directly from the parallelized C++ library it is depending on.

The pyscal module is an example of creating Python bindings for a C++ program, so that users can have the advantage of the flexibility of Python while retaining the performance of C++.

The second n2p2 module highlights that algorithmic changes can also significantly improve the performance of a software. In the present case the modifications of the core library lead to a substantially reduced memory footprint of all depending applications, which is crucial for the very consuming training program.

5 Outlook

The report of Deliverable 1.5 of E-CAM describes 9 software modules of WP1 in classical molecular dynamics. As described in the grant agreement, they are "in the area of classical molecular dynamics responding to requests of users." These modules cover several domains within the area of classical molecular dynamics. The specific modules were primarily selected based on direct user requests, or based on the needs of collaborations with industrial or academic research partners, or, for ESDW modules, based on the interests of the ESDW participants. In this way, these modules respond to the requests of users.

These modules include three based on OpenPathSampling, three based on OpenMM, two based on n2p2, and one that introduces its own package, pyscal. Overall, two modules were developed at ESDWs, and seven were developed by PDRAs.

The n2p2 module providing atomic environment descriptor analysis tools will be essential regarding future pilot project work on implementing a neural network potential based coarse-grained models. The memory reduction n2p2 module was initiated by user request and will have a positive effect on all future applications to come.

The OpenMM simulation setup modules are of a somewhat transverse nature. They use OpenMM, a software library that is designed for classical molecular dynamics. However, thanks to the flexibility of that library, it can also be used for mesoscale modelling, normally the focus of E-CAM WP4. The OpenMM plectoneme and OpenMM copolymer modules are examples where we have used the expertise that WP1 has with OpenMM to create tools that would also be of interest to WP4.

There are currently many modules still in progress for WP1. These include several that were developed as part of the ESDW "Topics in Classical MD", held in Lyon in April 2019, and the ESDW "Inverse Molecular Design & Inference: building a Molecular Foundry" held in Dublin in November 2019, which have not yet had their concluding follow-up workshops. Many of these in-progress modules continue research themes that have been established in previous modules, such as path sampling, free energy perturbation methods, and neural network potentials. An additional theme that has emerged is the development of Python bindings to existing code. Some such module E-CAM modules, for example, pyscal, came out of the 2018–2019 ESDW in Turin, where Python bindings was one of the main topics. However, it was also a topic of interest at the 2019–2020 ESDW in Lyon.

At this time, no additional ESDWs are expected for WP1. The fourth WP1 ESDW, held in Lyon in April 2019, will have its follow-up in February 2020. Participants from the fifth and final ESDW, held in Dublin in November 2019, are continuing development of their modules. The final WP1 software module deliverable will include modules from those ESDWs, as well as contributions from the three E-CAM PDRAs in WP1.

References

Acronyms Used

CECAM Centre Européen de Calcul Atomique et Moléculaire

ESDW Extended Software Development Workshop

OPS OpenPathSampling

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/
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
https://www.e-cam2020.eu/scientific-reports/ ... https://www.e-cam2020.eu/scientific-reports/
OpenMM ... https://openmm.org
OPS ... https://openpathsampling.org
n2p2 ... https://compphysvienna.github.io/n2p2/
Jupyter notebook ... http://jupyter.org

Page 2

E-CAM Deliverable D1.1 ... https://dx.doi.org/10.5281/zenodo.841694
2017–2018 ESDW in Leiden, Netherlands ... https://www.cecam.org/workshop-details/306
2018–2019 ESDW in Turin, Italy ... https://www.cecam.org/workshop-details/172

Page 3

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

Page 4

LAMMPS ... https://lammps.sandia.gov/

Page 5

Classical MD section of the E-CAM Library ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/index.html#readme-classical-md
readme.rst of Gromacs engine in OPS module. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/OpenPathSampling/ops_gromacs_engine/readme.html
Merge Request of Gromacs engine in OPS module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/11

Page 6

readme.rst of OPS Visit All States Ensemble module. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Mod
modules/OpenPathSampling/ops_visit_all_states/readme.html
Merge Request of OPS Visit All States Ensemble. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/146

Page 7

readme.rst of Interface-Constrained Shooting in OPS module. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/OpenPathSampling/ops_interface_shooting/readme.html
Merge Request of Interface-Constrained Shooting in OPS module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/148
OpenMMTools ... http://openmmtools.readthedocs.org
readme.rst of Double-Well Dimer Testsystems module. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/dw_dimer_testsystem/readme.html
Merge Request of Double-Well Dimer Testsystems module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/77

Page 8

histone ... https://en.wikipedia.org/wiki/Histone
Chromosome-Conformation-Capture ... https://en.wikipedia.org/wiki/Chromosome_conformation_capture
binders model ... https://www.ncbi.nlm.nih.gov/pubmed/22988072
HPC Dask ... https://dask.org/
Python compiler Numba ... http://numba.pydata.org/
OpenMM Copolymer module ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/index.html
Merge Request of OpenMM Copolymer module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/179

Page 9

linking number ... https://en.wikipedia.org/wiki/Linking_number
magnetic or optical tweezers ... https://en.wikipedia.org/wiki/Magnetic_tweezers
HPC Dask ... https://dask.org/
Python compiler Numba ... http://numba.pydata.org/
OpenMM Plectoneme module ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/index.html
Merge Request of OpenMM Plectoneme module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/178
pybind11 ... https://pybind11.readthedocs.io/en/stable/intro.html
BondOrderAnalysis ... https://homepage.univie.ac.at/wolfgang.lechner/bondorderparameter.html
Voro++ ... http://math.lbl.gov/voro++/

Page 10

readme.rst of pyscal module. ... https://e-cam.readthedocs.io/en/latest/Classical-MD-Modules/modules/pyscal/readme.html
Merge Request of pyscal module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/150
WP1 pilot project ... https://www.e-cam2020.eu/pilot-projects-with-industry/

Page 11

readme.rst of NNP-CG - Descriptor analysis module. ... https://gitlab.e-cam2020.eu/singraber/E-CAM-Library/blob/nnpcg_descriptor_analysis/Classical-MD-Modules/modules/nnpcg/nnpcg_descriptor_analysis/readme.rst
Merge Request of NNP-CG - Descriptor analysis module. ... https://gitlab.e-cam2020.eu/e-cam/E-CAM-Library/merge_requests/206
readme.rst of n2p2 - Symmetry Function Memory Footprint Reduction module. ... https://gitlab.e-cam2020.eu/singraber/E-CAM-Library/blob/n2p2_reduce_symfunc_memory/Classical-MD-Modules/modules/n2p2/n2p2_reduce_symfunc_memory/readme.rst
Merge Request of n2p2 - Symmetry Function Memory Footprint Reduction module. ... https://gitlab.e-cam2020.eu:10443/e-cam/E-CAM-Library/merge_requests/197

Page 13

ESDW "Topics in Classical MD" ... https://www.cecam.org/workshop-details/84
ESDW "Inverse Molecular Design & Inference: building a Molecular Foundry" ... https://www.cecam.org/workshop-details/81

---

[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] Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98(14):146401, April 2007. doi: 10.1103/PhysRevLett.98.146401.

[4] David W. H. Swenson, Jan-Hendrik Prinz, Frank Noe, John D. Chodera, and Peter G. Bolhuis. OpenPathSampling: A Python framework for path sampling simulations. 2. Building and customizing path ensembles and sample schemes. *Journal of Chemical Theory and Computation*, 15(2):837–856, 2019. doi: 10.1021/acs.jctc.8b00627. URL https://doi.org/10.1021/acs.jctc.8b00627.

[5] Peter Bolhuis. Rare events via multiple reaction channels sampled by path replica exchange. *The Journal of chemical physics*, 129:114108, 10 2008. doi: 10.1063/1.2976011.

[6] David W. H. Swenson and Peter G. Bolhuis. A replica exchange transition interface sampling method with multiple interface sets for investigating networks of rare events. *The Journal of Chemical Physics*, 141(4):044101, July 2014. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.4890037. URL http://dx.doi.org/10.1063/1.4890037.

[7] Jutta Rogal and Peter G. Bolhuis. Multiple state transition path sampling. *The Journal of Chemical Physics*, 129 (22):224107, December 2008. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.3029696. URL http://dx.doi.org/10.1063/1.3029696.

[8] Paul J. Steinhardt, David R. Nelson, and Marco Ronchetti. Bond-orientational order in liquids and glasses. *Phys. Rev. B*, 28(2):784–805, July 1983. ISSN 0163-1829. doi: 10.1103/physrevb.28.784. URL http://dx.doi.org/10.1103/physrevb.28.784.

[9] Walter Mickel, Sebastian C. Kapfer, Gerd E. Schröder-Turk, and Klaus Mecke. Shortcomings of the bond orientational order parameters for the analysis of disordered particulate matter. *The Journal of Chemical Physics*, 138 (4):044501, January 2013. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.4774084. URL http://dx.doi.org/10.1063/1.4774084.

[10] Stefan Auer and Daan Frenkel. Numerical simulation of crystal nucleation in colloids. In *Advanced Computer Simulation*, pages 149–208. Springer Berlin Heidelberg, January 2005. ISBN 9783540220589, 9783540315582. doi: 10.1007/b99429. URL http://dx.doi.org/10.1007/b99429.

[11] Wolfgang Lechner and Christoph Dellago. Accurate determination of crystal structures based on averaged local bond order parameters. *The Journal of Chemical Physics*, 129(11):114707, September 2008. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2977970. URL http://dx.doi.org/10.1063/1.2977970.

[12] Sarath Menon, Grisell Díaz Leines, and Jutta Rogal. pyscal: A python module for structural analysis of atomic environments. *Journal of Open Source Software*, 4(43), 2019. doi: 10.21105/joss.01824. URL https://doi.org/10.21105/joss.01824.

[13] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. DeePCG: Constructing coarse-grained models via deep neural networks. *The Journal of Chemical Physics*, 149(3):034101, July 2018. doi: 10.1063/1.5027645.

[14] S. T. John and Gábor Csányi. Many-Body Coarse-Grained Interactions Using Gaussian Approximation Potentials. *The Journal of Physical Chemistry B*, 121(48):10934–10949, December 2017. doi: 10.1021/acs.jpcb.7b09636.

[15] Andreas Singraber, Tobias Morawietz, Jörg Behler, and Christoph Dellago. Parallel Multistream Training of High-Dimensional Neural Network Potentials. *Journal of Chemical Theory and Computation*, 15(5):3075–3092, May 2019. doi: 10.1021/acs.jctc.8b01092.