

# Martignac: Computational workflows for reproducible, traceable, and composable coarse-grained Martini simulations

Tristan Bereau,<sup>1, a)</sup> Luis J. Walter,<sup>1</sup> and Joseph F. Rudzinski<sup>2</sup>

<sup>1)</sup> Institute for Theoretical Physics, Heidelberg University, 69120 Heidelberg, Germany

<sup>2)</sup> Physics Department and CSMB Adlershof, Humboldt-Universität zu Berlin, 12489 Berlin, Germany

(Dated: 25 September 2024)

Despite their wide use and far-reaching implications, molecular dynamics (MD) simulations suffer from a lack of both traceability and reproducibility. We introduce Martignac: computational workflows for the coarse-grained (CG) Martini force field. Martignac describes Martini CG MD simulations as an acyclic directed graph, providing the entire history of a simulation—from system preparation to property calculations. Martignac connects to NOMAD, such that all simulation data generated are automatically normalized and stored according to the FAIR principles. We present several prototypical Martini workflows, including system generation of simple liquids and bilayers, as well as free-energy calculations for solute solvation in homogeneous liquids and drug permeation in lipid bilayers. By connecting to the NOMAD database to automatically pull existing simulations and push any new simulation generated, Martignac contributes to improving the sustainability and reproducibility of molecular simulations.

## I. INTRODUCTION

Despite ever-increasing attention and community efforts for the last half century, molecular dynamics (MD) simulations remain poorly shared, deficiently reproducible, and often devoid of history or traceability. The sharing of MD data is made complex due to the fragmentation of hardware, software code, force fields, and simply the sheer diversity of systems of interest.<sup>1</sup> Efforts in this direction include the BioExcel Building Blocks (BioBB) library,<sup>2</sup> the COVID-19 Molecular Structure and Therapeutics Hub,<sup>3</sup> a general index of MD-simulation repositories found online (MDverse),<sup>4</sup> and the Simulation Foundry.<sup>5</sup> Yet, a recent document reiterated specific needs for the community, including persistent, indexed, and open access to MD data, metadata annotation, application programming interfaces (APIs) for data exchange, and comprehensive provenance information (i.e., history of the simulation).<sup>6</sup> Here we propose a concrete end-to-end solution for the popular coarse-grained (CG) Martini biomolecular force field.<sup>7,8</sup> We introduce *Martignac*, a workflow manager that automatically connects to an online database, avoids redundant calculations by downloading existing entries, runs missing simulations, and subsequently uploads them to enrich the database (Fig. 1). Martignac offers a traceable, composable, and reproducible framework for general-purpose and high-throughput CG Martini simulations.

To enable reproducibility and provenance information, computer simulations need a strict, specialized, and systematic *workflow*: a management system designed to orchestrate activities and organize resources into processes. Scientific workflows typically compose individual units of calculations as a directed acyclic graph (DAG).

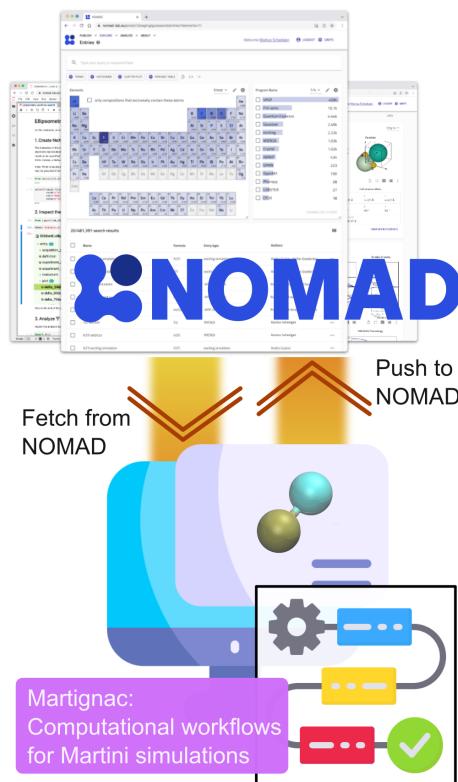


FIG. 1. Martignac implements Martini coarse-grained simulations as computational workflows. The library interacts with the NOMAD web server to automatically fetch any existing simulation, and push any new contribution.

They have rapidly become an essential ingredient to capture provenance information in materials modeling (e.g., see AiiDA workflows<sup>9</sup>). Martignac makes integral use of computational workflows. We will see that not only does this offer reproducible and traceable MD simulations, it also enables the *composability* of several MD

<sup>a)</sup> Electronic mail: bereau@uni-heidelberg.de

workflows together, thus avoiding redundant calculations upon compound screening.

Martignac implements workflows via `signac-flow`, a Python library that manages and automates workflows in computational research.<sup>10,11</sup> It organizes, executes, and monitors data processing pipelines, making it easier to handle complex and large-scale simulations and analyses. The framework has been applied to several projects, including the assembly of colloidal diamond,<sup>12</sup> photonic crystals,<sup>13</sup> and lubricating monolayer films.<sup>14,15</sup> The last application mentioned is the outcome of a larger consortium called MoSDeF, which offers a set of Python tools to help initialize, assign force-field parameters, and support screening of soft-matter systems.<sup>16</sup> Several of the authors of the last study have highlighted `signac-flow` as an essential tool to improve the reproducibility of molecular simulations, where they propose a set of principles to create Transparent, Reproducible, Usable by others, and Extensible (TRUE) molecular simulations.<sup>17</sup> Though not mentioned in the article, the TRUE principles overlap significantly with the much more widespread FAIR data principles: Findability, Accessibility, Interoperability, and Reuse of digital assets.<sup>18</sup>

The aspiration to work with FAIR data has seen rapid and intense developments in many areas of science.<sup>19–21</sup> In materials science, NOMAD<sup>22</sup> is one of the leading efforts in building FAIR databases.<sup>23–25</sup> Originally built as a repository for *ab initio* calculations, NOMAD has been recently transformed into a versatile research data management platform for a wide variety of materials science data.<sup>26</sup> Specifically relevant for this work, the openly-available NOMAD web-based platform provides the following functionalities: (*i*) automated detection and parsing of raw molecular simulation files from GROMACS, (*ii*) custom workflows that allow connections between independently run simulations and analysis stored in the database, and (*iii*) a full suite of API commands, enabling scriptable communication with the database. Martignac leverages these functionalities to not only facilitate transparent storage of the executed simulations and workflows but also to improve efficiency and prevent redundancy, i.e., to provide a comprehensive FAIR data management solution.

Martignac places a specific emphasis on high-throughput screening (HTS) applications. The Martini model has been an invaluable model for HTS applications due to its top-down parametrization: its building-block approach of CG bead types effectively reduces the size of chemical compound space.<sup>27,28</sup> The use of Martini for HTS has enabled a number of applications, including protein-ligand binding,<sup>29</sup> the extensive screening of drug-membrane permeation for 0.5 million compounds,<sup>30</sup> the potentials of mean force (PMFs) of all Martini dimers inserted in six phospholipid bilayers,<sup>31</sup> the identification of driving forces for generic anaesthetics,<sup>32</sup> and the molecular discovery of a molecular probe selective to cardiolipin.<sup>33</sup>

Martignac implements a select list of computational

workflows that is expected to be of general interest to the broader CG Martini community. We distinguish two categories of workflows: (*i*) system generation and (*ii*) free-energy calculation. The systems that Martignac can generate consist of: a solute in the gas (i.e., a molecule in an empty simulation box); a solvent (i.e., homogeneous fluid); a solute inserted in a solvent; and a phospholipid bilayer. The free-energy-calculation workflows comprise the solvation of a solute in a solvent, and the potential of mean force of a small molecule in a phospholipid bilayer. The reader may already foretell the workflow composability at play: for instance, calculating the solvation free energy of a solute requires to both generate the solute and the solvent independently, then combine them to generate the system, and finally run the free-energy calculations.

The present article first describes the NOMAD database in Sec. II, followed by Martignac's workflow methodology in Sec III, including the contents of the workflows and how they connect to NOMAD. In Sec. IV we summarize some of the MD simulation methods and parameters. Sec. V highlights a number of applications made possible by Martignac: how the content of the DAG generated by Martignac is translated to NOMAD, the composability of workflows, a reproducibility calculation of oil/water transfer free energies, and finally another reproducibility calculation for drug-membrane PMFs. We conclude with a number of final remarks and outlook in Sec. VI.

## II. NOMAD FUNCTIONALITIES

This section on the NOMAD database is only a short summary of the NOMAD documentation,<sup>34</sup> with a focus on aspects useful to Martignac.

### A. Processing and organization

NOMAD ingests the raw input and output files from standard simulation software by first identifying a representative file (e.g., the `log` file in the case of GROMACS) and then employing a parser code to extract relevant (meta)data from the associated files to that simulation. The (meta)data are stored within a structured schema—the NOMAD Metainfo—to provide context for each quantity, enabling interoperability and comparison between simulation software. The compilation of all (meta)data obtained from this processing forms an *entry*—the fundamental unit of storage within the NOMAD database—including simulation input/output, author information, and additional general overarching metadata (e.g., references or comments). In addition, a NOMAD entry offers unique identifiers: both an `entry_id` to manage unpublished data and also a DOI once published.

NOMAD entries can be organized hierarchically into *uploads*, *workflows*, and *datasets*. Since the parsing exe-

cution is dependent on automated identification of representative files, users are free to arbitrarily group simulations together upon upload. In this case, multiple entries will be created with the corresponding simulation data. An additional unique identifier, `upload_id`, will be provided for this group of entries. Although the grouping of entries into an upload is not necessarily scientifically meaningful, it is practically useful for submitting batches of files from multiple simulations to NOMAD. Concretely, Martignac utilizes uploads to group all  $\lambda$  coupling points of a thermodynamic-integration calculation. This is particularly convenient since NOMAD retains the original directory structure when storing all the raw and processed data.

NOMAD offers flexibility in the construction of workflows. First, a molecular dynamics simulation is a workflow in itself. The (meta)data for this standard workflow are automatically stored during processing and entail all relevant aspects: MD runtime parameters and ensemble-averaged or time-dependent observables. Furthermore, NOMAD also allows the creation of custom workflows, which are completely general directed graphs, allowing users to link NOMAD entries with one another in order to provide the *provenance* of the simulation data. Custom workflows are contained within their own entries and, thus, have their own set of unique identifiers. To create a custom workflow, the user is required to upload a workflow `yaml` file describing the inputs and outputs of each entry within the workflow, with respect to sections of the NOMAD Metainfo schema. Martignac automatically creates this file for its workflows, without the user being required to understand any details of the NOMAD schema.

At the highest level, NOMAD groups entries with the use of *datasets*. A NOMAD dataset allows the user to group a large number of entries, without any specification of links between individual entries. A DOI is also generated when a dataset is published, providing a convenient route for referencing all data used for a particular investigation within a publication.

## B. Programmatic access, query, and interaction

In addition to its GUI interface, NOMAD supports scriptable access to its database and functionalities through an extensive application programming interface (API).<sup>26</sup> A REST API queries the server by a combination of GET and POST requests. Martignac uses this API through a variety of python functions that lower the barrier for use by conveniently combining multiple API calls into a single routine and perform validation tests. In particular Martignac uses Marshmallow schemas to validate the input data that is received, as well as deserialize the input data to Python objects.<sup>35</sup> Read-only requests of publicly available entries can be made without NOMAD credentials. Otherwise, a NOMAD username and password is required to authenticate the API request.

## III. MARTIGNAC WORKFLOW METHODOLOGY

The present section focuses on the conceptual ideas behind Martignac. For a description of the Python library, we refer the reader to the online documentation.<sup>36</sup>

Workflows are built as directed acyclic graphs (DAGs). DAGs consist of nodes and single-directional edges. In our context, nodes are computational *operations*, e.g., generating a molecular configuration or running an MD simulation. These operations are interlinked—one cannot analyze a trajectory that has not yet been simulated—leading to a required ordering of the operations via directional connections, i.e., the DAG’s edges.

The Martignac implementation builds on the flexible `signac-flow` library.<sup>10,11</sup> `signac-flow` systematically manages and distributes a set of operations (e.g., MD simulations) repeated across many systems. This makes `signac-flow` appealing for high-throughput screening, where consistency is a paramount requirement. Each system studied is called a *state point*, which `signac-flow` associates to a unique and persistent job identifier (ID). Each job is thereby a collection of interdependent operations, each implemented as a Python function. In addition, the workflow requires label functions, which determine whether the operation has already been run. Finally, interdependences between operations (i.e., the DAG’s edges) are specified by pre- and post-conditions on other label functions.

Martignac workflows target various domain applications (e.g., generating a solvent or calculating the free energy of a solute in a phospholipid membrane). Because these domain applications share a number of operations, we separate workflows into two components:

1. Generic operations that we apply irrespective of the domain application;
2. Domain-application specific operations;

as shown in Fig. 2.

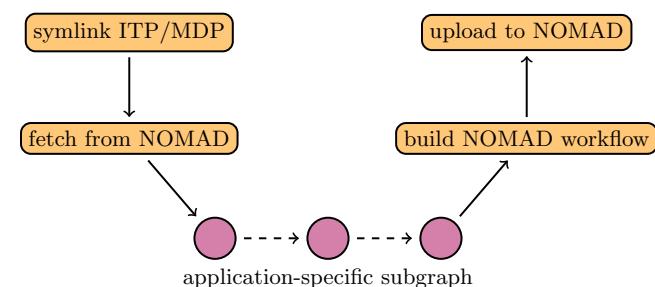


FIG. 2. The Martignac workflow structure is split in two subgraphs: (i) generic operations common to all simulation workflows (yellow); (ii) domain-application specific operations (see Fig. 4).

## A. Generic subgraph

The generic operations consist of the following:

- **symlink ITP/MDP:** Force-field-definition files are extremely redundant: the same sets of `*.itp` files are necessary to run any Martini simulation. Similarly, high-throughput workflows will typically work with identical `*.mdp` input-parameter files to obtain systematic simulations. A systematic upload of these files creates an unnecessary burden on storage requirements. To this end, Martignac makes use of symbolic links to mitigate local storage footprint and avoids uploading said files to the server. However, reproducibility remains: standard Martini force-field files are available online.<sup>37</sup> While input `*.mdp` parameter files are not saved, Martignac does store the (equivalent) output `*.mdp` files generated by the GROMACS preprocessor, `grompp`.
- **fetch from NOMAD:** Martignac queries the user's NOMAD dataset to look for an existing simulation already stored online. Martignac checks whether the simulation queried *exactly* corresponds to the one to be attempted. Comparison is made on the basis of (*i*) the workflow Python-class name, (*ii*) the `signac-flow` job ID, and (*iii*) a hash of all input `*.mdp` files involved. The information is contained as a JSON-formatted comment of the simulation upload, which is automatically generated and pushed with any upload (see below). For instance, Fig. 3 is a comment for a simulation upload of a single solute generation. This check ensures integrity beyond the mere desired workflow and chemistry, but also in the exact input files used.
- **build NOMAD workflow:** The chain of operations implemented in Martignac form a DAG. Said DAG is converted and serialized into a NOMAD-compatible workflow `yaml` file, for subsequent simulation upload.
- **upload to NOMAD:** All files generated during a Martignac workflow (except for the `*.itp` Martini definition files and input `*.mdps`) are zipped together with the `yaml` NOMAD workflow file and uploaded to the NOMAD webserver via an API POST request. A comment is attached to every upload containing a JSON-formatted string containing identifiable information about the content of the job.

The generic operations are arranged as shown in Fig. 2: a linear chain of operations with the application-specific subgraph in the middle.

```
{
  'job_id': '3e793a7b2a1e83233c40458fddf958ab',
  'itp_files': 'a52590b1d87d122ba1e376b83c3d6bee',
  ,
  'mdp_files': '2d7a9e52d14d23e0dfb97192d75a3463',
  ,
  'state_point': {
    'solute_name': 'P5', 'type': 'solute'
  },
  'workflow_name': 'SoluteGenFlow'}
```

FIG. 3. Comment of an example job upload. The respective keys correspond to the `signac-flow` job ID, the hashes of the collection of `*.itp` and `*.mdp` files, respectively, state-point dependent information, including the solute name and type, and finally the workflow Python-class name.

## B. Application-specific subgraphs

Here we shortly describe the directed acyclic subgraphs that are application specific, and contained within the larger Martignac DAG (see Fig. 2). This implies that all application-specific subgraphs described below are both preceded and followed by the node operations described in the generic subgraph above.

### 1. Solute generation

Generating a solute in the gas phase (i.e., an empty box, Fig. 4a) consists of three steps

1. **build:** From the name of the solute molecule, generate a structure, particle-definition, and topology files, `gro`, `itp`, and `top`, respectively.
2. **minimize:** Energy minimization.
3. **equilibrate:** Equilibration MD simulation.

The only state-point parameter is the name of the solute.

### 2. Solvent generation

We consider the generation of a homogeneous liquid that fills the simulation box (Fig. 4b)

1. **generate solvent molecule:** Generate a single solvent molecule. This step is analogous to the **build** part of the solute-generation workflow.
2. **build solvent box:** build a box of solvent molecules by means of the PACKMOL program.<sup>38</sup>
3. **convert box to gro:** The preceding PACKMOL operation yields a `pdb` file. The present operation simply converts the `pdb` to a `gro` structure file.
4. **minimize:** Energy minimization.

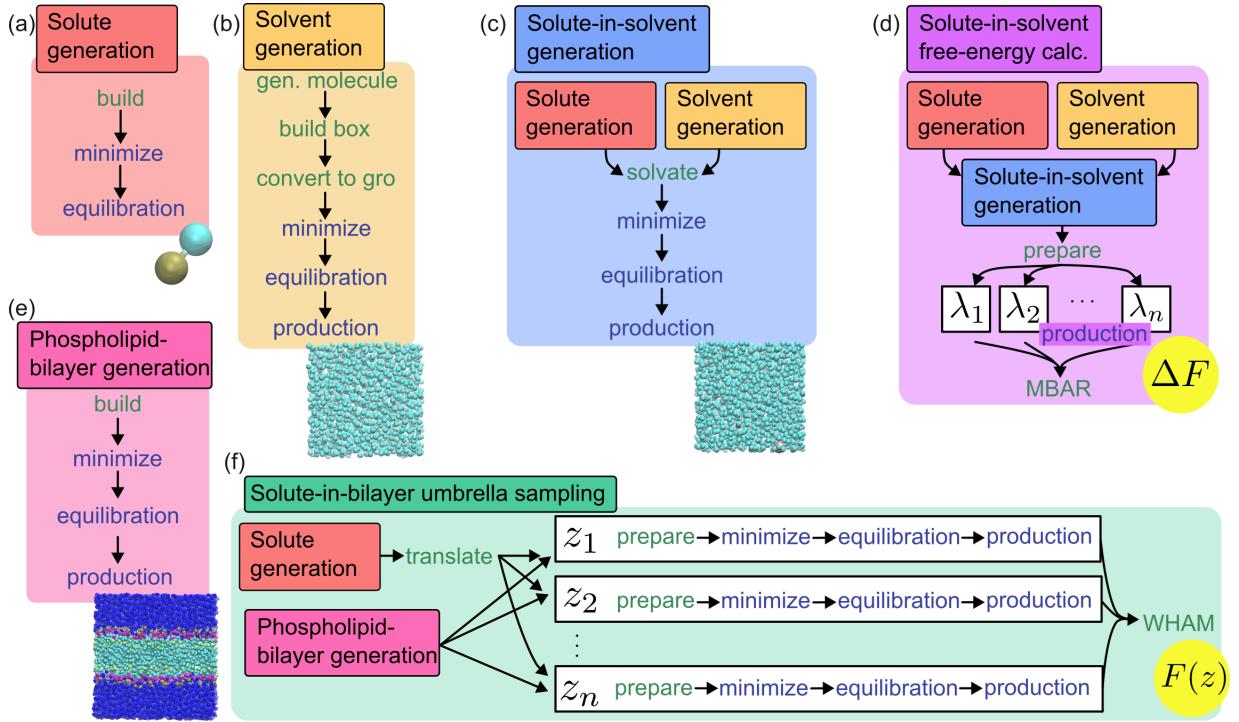


FIG. 4. Martignac workflows. System generation: (a) Solute; (b) solvent; (c) solute in solvent; (e) phospholipid bilayer. Free-energy calculations: (d) solute-in-solvent free energy; (f) solute-in-bilayer potential of mean force. Blue and green operations distinguish MD simulations from the others, respectively. The composability of workflows is highlighted by entire graphs being part of others. Aggregation of  $\lambda$  states or umbrella collective variables,  $z$ , recycles common operations and makes use of all relevant information for analysis. The outcome of each workflow is illustrated: either a generated system or a free energy.

5. **equilibrate**: Equilibration MD simulation. For instance, this could be run at constant pressure with a forgiving barostat, such as Berendsen or C-rescale.

6. **production**: Production MD simulation.

The only state-point parameter is the name of the solvent molecule.

### 3. Solute-in-solvent generation

The solute-in-solvent generation (Fig. 4c) makes use of the *composability* of the Martignac workflows: it first generates the solute and the solvent using the above-mentioned workflows, and subsequently joins them to yield the desired system. As such the DAG is not linear, but contains branches to link the individual components together.

1. **generate solute**: fetch or run the solute generation workflow (Sec. III B 1).
2. **generate solvent**: fetch or run the solvent generation workflow (Sec. III B 2).
3. **solvate**: solvate the solute using the GROMACS `gmx solvate` program.

4. **minimize**: Energy minimization.

5. **equilibrate**: Equilibration MD simulation.

6. **production**: Production MD simulation.

The state-point parameters are the solute and solvent names.

### 4. Solute-in-solvent alchemical transformation

Composability is leveraged once again here (Fig. 4d). We make use of the solute-in-solvent system generated in the last workflow. Free-energy calculations are employed to compute the free energy of coupling the solute in the solvent. We make use of thermodynamic integration, where a series of Hamiltonians interpolating between the two end states, denoted by the parameter  $\lambda \in [0, 1]$ , increasingly couple the solute in the simulation box. As such the DAG needs to be run not only once for the system of interest, but  $n$  times, indicative of the number of interpolating Hamiltonians. The present workflow DAG is thereby *nonlinear*: it will run MD simulations for each  $\lambda$  state in parallel. However, both the system initialization and the final free-energy calculation ought to occur only once. This is illustrated in Fig. 4 (d). Concretely, this is implemented by means of an *aggregator* function decorator.

1. **prepare system:** Fetch or run the solute generation, solvent generation, and solute-in-solvent generation workflows. (Aggregated operation.)
2. **production:** Production MD simulation at a specific  $\lambda$  value, additionally evaluating and storing the energy from all interpolating Hamiltonians,  $U_\lambda$ , for later use in the free-energy calculations.
3. **compute free energy:** Compute the free energy by means of the multi-Bennett acceptance ratio (MBAR)<sup>39</sup> via the `alchemlyb` library.<sup>40</sup> (Aggregated operation.)

The state-point parameters are the solute name, solvent name, and Hamiltonian-coupling  $\lambda$  value.

## 5. Phospholipid-bilayer generation

Here we consider the generation of a phospholipid bilayer (Fig. 4e). The implementation not only allows for a variety of single-composition (Martini-supported) lipid bilayers, it also supports the generation of lipid mixtures.

1. **generate initial bilayer:** Generation of an initial phospholipid-bilayer structure by means of the INSANE tool.<sup>41</sup>
2. **minimize** Energy minimization.
3. **equilibrate** Equilibration MD simulation.
4. **production** Production MD simulation.

The relevant state-point parameters are the name and fractional composition of each phospholipid name.

## 6. Solute-in-bilayer umbrella sampling

We consider the thermodynamics of insertion of a solute molecule in a phospholipid bilayer (Fig. 4f). We compute the potential of mean force (PMF) by means of umbrella sampling.<sup>42</sup> We consider the PMF of insertion against a typical collective variable: the depth normal to the bilayer,  $z$ . This last workflow is again a combination of composability and state-point aggregation: Composability of the solute generation (Sec. III B 1) with phospholipid-bilayer generation (Sec. III B 5); and state-point aggregation when collecting umbrella-sampling restraints placed at various intervals of the collective variable,  $z$ .

1. **generate solute:** fetch or run the solute generation workflow (Sec. III B 1). (Aggregated operation.)
2. **translate solute:** move the solute to the origin of the simulation box. (Aggregated operation.)

3. **generate bilayer:** fetch or run the phospholipid-bilayer generation workflow (Sec. III B 5). (Aggregated operation.)
4. **insert solute in box:** use PACKMOL to place the solute in the bilayer box.
5. **convert box to gro:** Convert PACKMOL's output `pdb` file to `gro` format.
6. **update topology file:** Combine the topology files of the solute and bilayer systems.
7. **minimize** energy minimization.
8. **equilibrate** MD-based equilibration simulation.
9. **production** Production MD simulation.

10. **compute WHAM:** Use GROMACS' implementation of the weighted histogram analysis method (WHAM) to compute the PMF.<sup>43</sup> (Aggregated function.)
11. **analyze WHAM:** Convert and store the GROMACS WHAM output `xvg` files to `numpy` arrays.

## IV. SIMULATION METHODS

Molecular dynamics (MD) simulations were performed with GROMACS 2023.1.<sup>44</sup> Unless specified, we relied on the Martini 3 force-field parameters<sup>8</sup> with an integration time step of  $\delta t = 0.02 \tau$ , where  $\tau$  is the model's natural unit of time. Simulations targeted an *NPT* ensemble: constant number of particles, pressure ( $P = 1$  bar), and temperature ( $T = 298$  K). The latter was controlled by means of a stochastic velocity-rescaling thermostat.<sup>45</sup> Equilibration MD simulations typically made use of the Berendsen or C-rescale barostats, while production simulations relied on the more accurate, but also more sensitive, Parrinello-Rahman barostat.<sup>46</sup>

To generate solvent boxes, we used the PACKMOL program,<sup>38</sup> and INSANE was used to generate phospholipid bilayers.<sup>41</sup> Various tools of the GROMACS suite were used to generate `gro` structures and topology files, solvate a solute, or run the weighted histogram analysis method (WHAM). Alchemical free energies were calculated by means of the multi-Bennett acceptance ratio (MBAR)<sup>39</sup> via the `alchemlyb` library.<sup>40</sup>

Because of variations in the exact protocol used in the various workflows, we refer the reader to the Martignac implementation or published NOMAD entries for more detailed information. In particular, the full set of parsed simulation input parameters can be easily browsed via NOMAD's MetaInfo viewer, found under the "Data" tab of each entry page.

## V. RESULTS

This section highlights a number of features enabled by Martignac’s computational-workflow design. To accompany the results, we systematically refer to the hyperlinked NOMAD upload ID for easy access to each computational workflow and underlying simulations. We also provide a graphical user interface to the NOMAD uploaded Martignac uploads in a web-based app on Streamlit.<sup>47</sup> The Streamlit app dynamically queries NOMAD, and features application-specific properties, such as the underlying DAG, free energies, and potentials of mean force.

### A. The Martignac directed graph translates to NOMAD metadata

As a first example of the interaction between Martignac and NOMAD, we consider the generation of a box of hexadecane molecules—one of the standard Martini solvents. Fig. 5 (a) shows a DAG that is automatically generated by reading Martignac’s set of operations and pre/post-conditions. The DAG contains all generic and application-specific operations. Though this DAG is linear, others in this work have non-trivial connectivity, owing to loading several other workflows as part of the early system setup, or aggregation of simulations for free-energy calculations.

In comparison to the Martignac DAG, we also show in Fig. 5 (b) an illustration of the workflow that is generated and interpreted by NOMAD. NOMAD correctly identifies all operations, and even distinguishes operations that consist of MD simulations from the others. NOMAD’s correct visualization of the workflow validates the programmatic transfer of the DAG into simulation metadata. An example can be found for the (hyperlinked) NOMAD upload ID [uzssztc-SrGcz49GuSVlqQ](#).

### B. Workflow composability

Avoiding unnecessary redundancies is an important feature of high-throughput calculations, because they can save significant compute and storage resources. For instance, the screening of the insertion of a solute in a solvent involves compounding combinatorics: each solute against each solvent. We avoid generating the same system twice by enforcing composability. Several workflows start with the generation of the elementary parts, e.g., solute and solvent in a solute-in-solvent system, or solute and lipid bilayer in drug permeation. Each elementary step relies on a “fetch-or-run” mechanism: we first check whether the system has been previously run by means of an API call. If so, we download it, otherwise, we run the system. If downloaded, Martignac locally stores the NOMAD upload ID of the elementary workflow. The elementary workflow is included in the final NOMAD

workflow by referencing the elementary upload ID. This referencing of existing workflows enables a hierarchical structure and reusability of simulations. We check that when running a high-throughput calculation of solute-in-a-solvent generation, a given chemistry for a solute is ever calculated and stored only once. Case in point: the alchemical calculation for the solute bead P6 in a homogeneous liquid of hexadecane is stored in a single upload with (hyperlinked) ID [PCQSjL2wQsCptzZhHa196Q](#). Every  $\lambda$ -point simulation relies on the same solute system generation [oT0F5qP9RY6AFDiDQrJ2ug](#), which is built from a single solute generation, [7YU8feV.SQ6h6M7aIUCdQg](#), and a single solvent generation, [uzssztc-SrGcz49GuSVlqQ](#).

### C. Reproducibility of oil/water transfer free energies

Martignac facilitates reproducibility by the systematic nature of its computational workflows. As a first example, we focus on oil/water transfer free energies. The recent Martini 3 force field provides an extensive reference of free-energy calculations as supporting information.<sup>8</sup> Parts of these reference thermodynamic calculations include oil/water transfer free energies for the majority of CG beads defined by the Martini model. Here we reproduce a subset of these calculations by means of the *Solute-in-solvent alchemical transformation* workflow (Fig. 4d). Because the workflow incorporates not only system generation and MD simulations, but also the calculations of the free energies themselves, these are straightforward to store as metadata in NOMAD. As such, we directly query the free energies from NOMAD to fetch easily- and permanently-available thermodynamic properties.

Solute	Solvent	Martignac	Reference
P6	HD → W	-27.45	-27.20
	CLF → W	-11.98	-11.90
	ETH → W	-10.96	-11.20
	CHEX → W	-18.86	-19.00
	W	17.98	18.00

TABLE I. Reproducibility of solute-in-solvent free energies against the Martini 3 publication.<sup>8</sup> Solvents with and without a right-pointing arrow denote transfer and hydration free energies, respectively. All free energies in units of kJ/mol.

Tab. I shows a comparison of the free energies we obtain from Martignac, and the reference values from the Martini 3 study. Though the hydration free energy (i.e., solvation in water) is readily calculated from Martignac, the other fields consist of transfer free energies from oil to water. These are simply computed by subtracting the two individual solvation free energies. All values are in excellent agreement of one another, within 0.3 kJ/mol for each one of them. To further demonstrate the ability to fetch the free energies from the data directly, we refer the reader to our Streamlit app, which fetches metadata from NOMAD to display the free energy resulting from each computational workflow.<sup>47</sup>

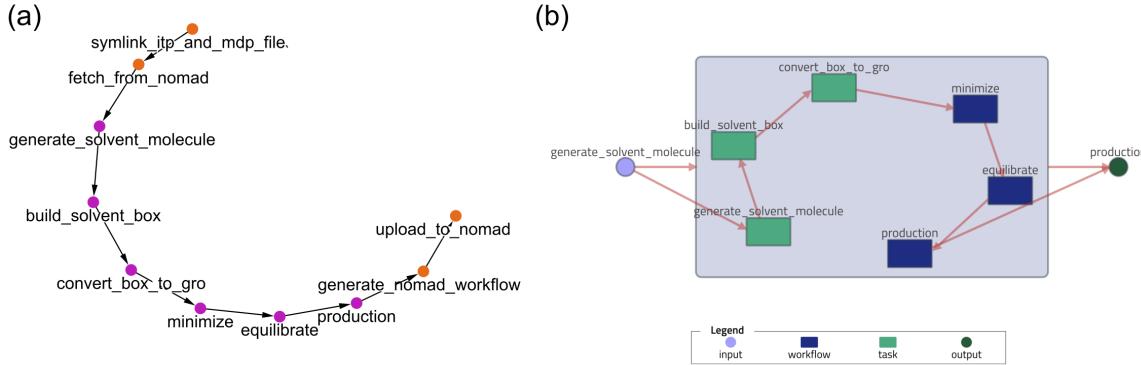


FIG. 5. Workflow graph for solvent generation. (a) Workflow generated from Martignac. Color coding follows Fig. 2. (b) Workflow generated from NOMAD. Green and dark-blue rectangles display operations, and distinguish those that involve an MD simulation.

The NOMAD upload IDs for the alchemical calculations of P6 in the solvents HD, CLF, ETH, CHEX, and W are, respectively: PCQSjL2wQsCptZhHal96Q, GAugbmarSJmlADE8rHK4AQ, HKHmU0bpQ\_a-SwaMVxvSeQ, w4sPShVOSTm9Yc-bVuFvYw, and ZwcN37wMSyidNQeQPBQbAw.

#### D. Reproducibility of drug-membrane potentials of mean force

As a second example of reproducibility, we consider the potentials of mean force (PMFs) of small Martini molecules inserted into a phospholipid bilayer. We perform PMF calculations for the C1-P4 dimer and SC1-SP2-SC1 trimer in a 1-palmitoyl-2-oleyl-*sn*-glycero-3-phosphocholine (POPC) bilayer, found originally in Hoffmann et al.<sup>31</sup> and Potter et al.,<sup>48</sup> respectively. Both studies rely on the Martini 2 force field.<sup>7,49,50</sup> The PMFs are calculated using the *Solute-in-bilayer umbrella sampling* workflow (Fig. 4f). Fig. 6 shows the original and Martignac PMFs in dashed and solid lines, respectively. The PMFs of the C1-P4 dimer show excellent agreement, with only a slight deviation around the first bead of the lipid tail region. As Hoffmann et al. provided simulation input files, reproducing the simulations was straightforward. The update to a more recent version of GROMACS together with statistical convergence likely explain the slight variations between the PMFs. For the SC1-SP2-SC1 trimer, the Martignac PMF generally matches the result of the original study, but is slightly shifted down due to deviations in the water region ( $z \gtrsim 2.6$  nm) used as the zero reference. Although Potter et al. do not provide simulation input files, they include all essential parameters in their method description. We utilized their specified parameters in conjunction with defaults for the unspecified parameters. However, in contrast to extracting parameters from the method description, providing a complete input file facilitates the reproduction of a simulation and generally

prevents missing parameter information. Martignac uploads all simulation input and output files pertaining to each PMF calculation as a single upload to NOMAD. As the workflow also incorporates WHAM calculations, the resulting PMF curves are included in the NOMAD upload. For the dimer and the trimer, the corresponding NOMAD upload IDs are VpbwP6VpS4ucuwqVBHPzeg and N51F6fmXR16BHpNEQNBPtQ, respectively.

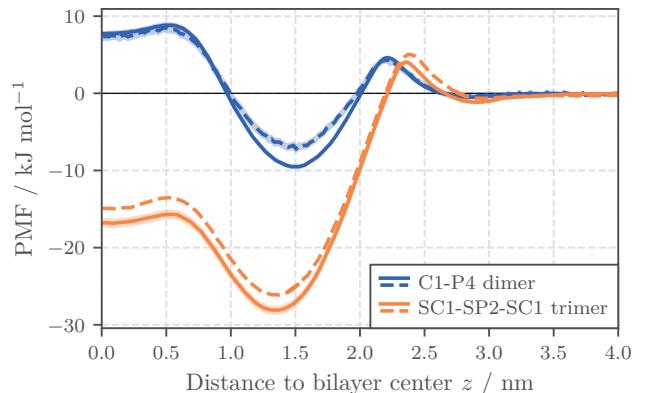


FIG. 6. Drug-membrane potentials of mean force (PMFs) for a C1-P4 dimer and an SC1-SP2-SC1 trimer inserted in a POPC bilayer. Solid and dashed lines correspond to Martignac and the original studies,<sup>31,48</sup>, respectively. Error estimates from bootstrapping (not available for the original trimer result) are shown as (relatively small) shaded areas.

We now consider the reproducibility of polyethylene, mapped to a C1-C1 dimer, in a POPC membrane from Bochicchio et al.<sup>51</sup> As the original study did not provide all essential simulation parameters, we could not precisely reproduce the simulation setup. While relevant parameters for the umbrella sampling, the temperature coupling, and the barostat are specified, information about the treatment of non-bonded interactions is missing. We investigate the impact of different methods for handling electrostatic and van der Waals (VdW)

interactions on the resulting PMFs. Fig. 7 shows the result from the original study together with five different variants obtained with the Martignac workflow. The NOMAD upload IDs are ordered from top to bottom: `UkYrTZTDRUGOrTpsuJzd7Q`, `Eiatr72XQu6X03u0w6T15A`, `4x-JWH_IQIqqHevHf5LABQ`, `owgFZkssRd6kttr1KitsTQ`, and `XmFAOFG1Q3qd7KDBsF9mCw`. While various methods for handling electrostatic interactions do not significantly impact the PMF curve, the VdW treatment causes greater differences in our results. Despite testing multiple parameters for treating non-bonded interactions, we unfortunately could not reproduce the result from Bochicchio et al. Notably, our PMF from simulations using a VdW cutoff are in excellent agreement with the C1–C1 dimer results from Hoffmann et al.<sup>31</sup> Additionally, our PMF more closely resembles the atomistic calculation provided as part of the original study.<sup>51</sup> In particular, the PMF peak near the membrane–water interface is closer to the bilayer center for their atomistic result, aligning more closely with our findings. In general, the discrepancies observed may be attributed to factors such as unsatisfactory simulation convergence or further differences in the simulation setup. For instance, the use of the polarizable water model might shift the membrane thickness, but it is unfortunately not supported by Martignac at the moment. Overall this makes a strong point for the broad and systematic use of FAIR data storage for molecular simulations.

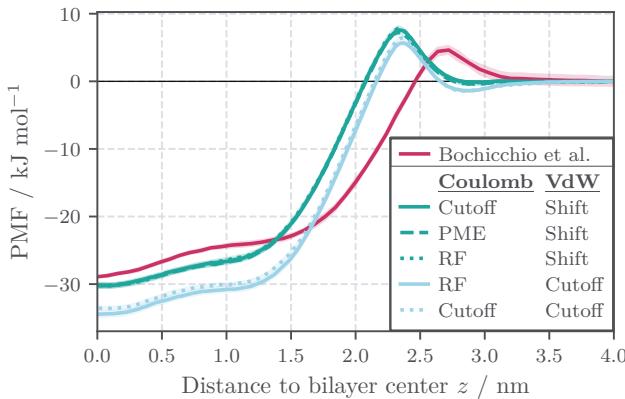


FIG. 7. Comparison of multiple computations of drug-membrane potentials of mean force (PMFs) for a C1–C1 dimer inserted in a POPC bilayer with results reported by Bochicchio et al.<sup>51</sup> We employed various combinations of methods to address electrostatic (Coulomb) and van der Waals (VdW) interactions as implemented in GROMACS. Results using the same VdW interaction handling overlap almost entirely due to excellent agreement. We show error estimates from bootstrapping as (relatively small) shaded areas for our simulation results.

A notable benefit of storing simulation workflows online is the ability to both query and display scientific information in a user-friendly fashion—here illustrated with the Martignac Streamlit app.<sup>47</sup> The app queries a

NOMAD database to list all systems corresponding to a Martignac workflow. In Fig. 8, we show the solute-in-bilayer umbrella sampling subpage of the app. The top part of the figure displays the Martignac DAG, highlighting the branching upon system generation. Further, the bottom part shows a list of systems found in the database. For each, various information extracted from the NOMAD entry are reported. Notably, this enables us to report and display simulation outputs: here the app systematically constructs the PMF of the workflow. Such online, interactive, and visually appealing aspects are likely to promote FAIR molecular-simulation data storage.

## Martignac

### Solute-in-bilayer umbrella sampling

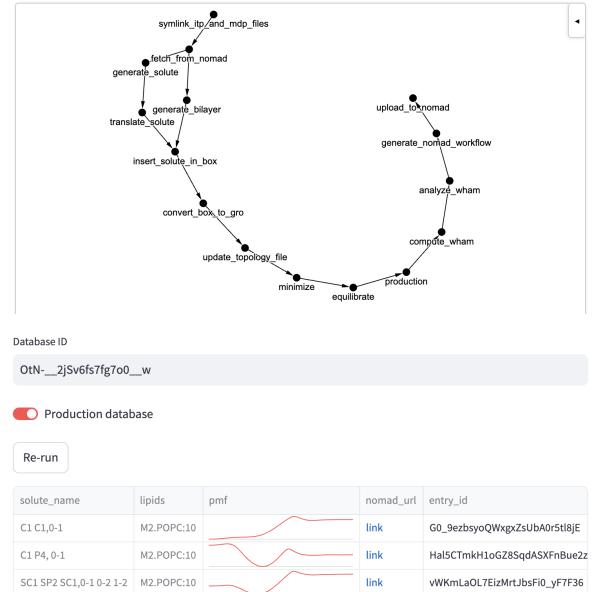


FIG. 8. Snapshot of the Streamlit web app. Top: Illustration of the solute-in-bilayer umbrella sampling. Bottom: systems found in the NOMAD database `0tN--2jSv6fs7fg7o0_w`. The query fetches the results of the WHAM calculations to automatically display the PMFs in small format.

## VI. CONCLUSION

We introduce Martignac: computational workflows for the coarse-grained (CG) Martini biomolecular force field. Legacy Bash scripts with error-prone copy/pasting make way for a more robust approach by means of workflows. The set of operations relevant to a particular objective (e.g., generating a box of solvent or calculating a free energy) are connected in an acyclic directed graph (DAG). The DAG links said operations to offer a *traceable history* from system generation, to MD simulations, to analysis and estimation of material/thermodynamic properties. The history offers anyone the ability to inspect,

check, and reproduce the content at each step. Moreover, the definition of elementary workflows enables their composability: separating system generation from further analysis means that a single instance of the former can be applied to a variety of downstream calculations. Here, we not only separate system generation from free-energy calculations, we split system generation in terms of their basic components: solutes, homogeneous liquids, and lipid bilayers. We show that Martignac greatly facilitates both reproducibility and composability by means of several examples pertaining to oil/water transfer free energies and drug-membrane thermodynamics.

The deep interconnection between Martignac and NOMAD carries interesting benefits. First, the systematic pulling of existing workflows greatly improves sustainability: the community can download existing Martini simulations, instead of simulating them (again). The automatic *pushing* of missing workflows removes any friction or efforts associated with publishing MD simulations. In this way, the user helps the community by enriching the corpus of Martini simulations available online. Finally, we find that publishing entire computational workflows offers a solution to the recent increase in the volume of scientific articles' supplementary information: all relevant data and metadata is stored and accessible in the NOMAD entries.

The connection to NOMAD also means that all simulation metadata is persistently available online. We refer the reader to the Martignac Streamlit app.<sup>47</sup> The app fetches all published Martignac simulations. The connection to the NOMAD API means that the entries are constantly updated with added simulations. Similar to MDverse, the app offers an intuitive user interface to browse through simulations. The added benefit of Martignac is the access to the simulation metadata, allowing us to automatically sort between workflows and extract scientifically meaningful information, such as free energies. Looking ahead, the incorporation of workflows for more biomolecular simulations of interest is straightforward, and will further help move the field to more systematic practices.

## ACKNOWLEDGMENTS

We thank Brandon Butler and Corwin Kerr for discussion about the `signac` workflow library. T.B. acknowledges support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster). J.F.R.'s contribution was funded by the NFDI consortium FAIRmat - Deutsche Forschungsgemeinschaft (DFG) - Project 460197019. Icons on figure 1 made by Freepik and Haca Studio from [www.flaticon.com](http://www.flaticon.com).

<sup>1</sup>M. Abraham, R. Apostolov, J. Barnoud, P. Bauer, C. Blau, A. M. Bonvin, M. Chavent, J. Chodera, K. Čondić Jurkić, L. Dele-

motte, *et al.*, "Sharing data from molecular simulations," *Journal of chemical information and modeling* **59**, 4093–4099 (2019).

<sup>2</sup>P. Andrio, A. Hospital, J. Conejero, L. Jordá, M. Del Pino, L. Codo, S. Soiland-Reyes, C. Goble, D. Lezzi, R. M. Badia, *et al.*, "Bioexcel building blocks, a software library for interoperable biomolecular simulation workflows," *Scientific data* **6**, 169 (2019).

<sup>3</sup>The Molecular Sciences Software Institute (MolSSI) and BioExcel, "COVID-19 Molecular Structure and Therapeutics Hub," <https://covid.molssi.org/>, accessed: 2024-08-06.

<sup>4</sup>J. K. Tiemann, M. Szczuka, L. Bouarroudj, M. Oussaren, S. Garcia, R. J. Howard, L. Delemotte, E. Lindahl, M. Baaden, K. Lindorff-Larsen, *et al.*, "Mdverse: Shedding light on the dark matter of molecular dynamics simulations," *eLife* **12** (2023).

<sup>5</sup>G. Gygli and J. Pleiss, "Simulation foundry: Automated and fair molecular modeling," *Journal of chemical information and modeling* **60**, 1922–1927 (2020).

<sup>6</sup>R. Amaro, J. Åqvist, I. Bahar, F. Battistini, A. Bellaiche, D. Beltran, P. C. Biggin, M. Bonomi, G. R. Bowman, R. Bryce, *et al.*, "The need to implement fair principles in biomolecular simulations," *arXiv preprint arXiv:2407.16584* (2024).

<sup>7</sup>S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. De Vries, "The martini force field: coarse grained model for biomolecular simulations," *The journal of physical chemistry B* **111**, 7812–7824 (2007).

<sup>8</sup>P. C. Souza, R. Alessandri, J. Barnoud, S. Thallmair, I. Faustino, F. Grünewald, I. Patmanidis, H. Abdisadeh, B. M. Bruininks, T. A. Wassenaar, *et al.*, "Martini 3: a general purpose force field for coarse-grained molecular dynamics," *Nature methods* **18**, 382–388 (2021).

<sup>9</sup>M. Uhrin, S. P. Huber, J. Yu, N. Marzari, and G. Pizzi, "Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows," *Computational Materials Science* **187**, 110086 (2021).

<sup>10</sup>C. S. Adorf, P. M. Dodd, V. Ramasubramani, and S. C. Glotzer, "Simple data and workflow management with the signac framework," *Computational Materials Science* **146**, 220–229 (2018).

<sup>11</sup>V. Ramasubramani, C. Adorf, P. Dodd, B. Dice, and S. Glotzer, "signac: A python framework for data and workflow management," in *Proceedings of the Python in Science Conference* (2018).

<sup>12</sup>Y. Zhou, R. K. Cersonsky, and S. C. Glotzer, "A route to hierarchical assembly of colloidal diamond," *Soft Matter* **18**, 304–311 (2022).

<sup>13</sup>R. K. Cersonsky, J. Antonaglia, B. D. Dice, and S. C. Glotzer, "The diversity of three-dimensional photonic crystals," *Nature communications* **12**, 2543 (2021).

<sup>14</sup>A. Z. Summers, J. B. Gilmer, C. R. Iacovella, P. T. Cummings, and C. McCabe, "Mosdef, a python framework enabling large-scale computational screening of soft matter: Application to chemistry-property relationships in lubricating monolayer films," *Journal of Chemical Theory and Computation* **16**, 1779–1793 (2020).

<sup>15</sup>C. D. Quach, J. B. Gilmer, D. Pert, A. Mason-Hogans, C. R. Iacovella, P. T. Cummings, and C. McCabe, "High-throughput screening of tribological properties of monolayer films using molecular dynamics and machine learning," *The Journal of Chemical Physics* **156** (2022).

<sup>16</sup>MoSDeF, "Molecular Simulation Design Framework (MoSDeF)," <https://mosdef.org/index.html>, accessed: 2024-08-06.

<sup>17</sup>M. W. Thompson, J. B. Gilmer, R. A. Matsumoto, C. D. Quach, P. Shamaprasad, A. H. Yang, C. R. Iacovella, C. McCabe, and P. T. Cummings, "Towards molecular simulations that are transparent, reproducible, usable by others, and extensible (true)," *Molecular physics* **118**, e1742938 (2020).

<sup>18</sup>M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data* **3**, 1–9 (2016).

- <sup>19</sup>T. Tanhua, S. Pouliquen, J. Hausman, K. O'brien, P. Bricher, T. De Bruin, J. J. Buck, E. F. Burger, T. Carval, K. S. Casey, *et al.*, “Ocean fair data services,” *Frontiers in Marine Science* **6**, 440 (2019).
- <sup>20</sup>N. Jeliazkova, M. D. Apostolova, C. Andreoli, F. Barone, A. Barrick, C. Battistelli, C. Bossa, A. Botea-Petcu, A. Châtel, I. De Angelis, *et al.*, “Towards fair nanosafety data,” *Nature Nanotechnology* **16**, 644–654 (2021).
- <sup>21</sup>T. J. Jacobsson, A. Hultqvist, A. García-Fernández, A. Anand, A. Al-Ashouri, A. Hagfeldt, A. Crovetto, A. Abate, A. G. Ricciardulli, A. Vijayan, *et al.*, “An open-access database and analysis tool for perovskite solar cells based on the fair data principles,” *Nature Energy* **7**, 107–115 (2022).
- <sup>22</sup>NOMAD, “NOMAD Homepage,” <https://nomad-lab.eu/> (), accessed: 2024-08-06.
- <sup>23</sup>C. Draxl and M. Scheffler, “Nomad: The fair concept for big data-driven materials science,” *Mrs Bulletin* **43**, 676–682 (2018).
- <sup>24</sup>C. Draxl and M. Scheffler, “The nomad laboratory: from data sharing to artificial intelligence,” *Journal of Physics: Materials* **2**, 036001 (2019).
- <sup>25</sup>M. Scheffler, M. Aeschlimann, M. Albrecht, T. Bereau, H.-J. Bungartz, C. Felser, M. Greiner, A. Groß, C. T. Koch, K. Kremer, *et al.*, “Fair data enabling new horizons for materials research,” *Nature* **604**, 635–642 (2022).
- <sup>26</sup>M. Scheidgen, L. Himanen, A. N. Ladines, D. Sikter, M. Nakhaee, Á. Fekete, T. Chang, A. Golparvar, J. A. Márquez, S. Brockhauser, *et al.*, “Nomad: A distributed web-based platform for managing materials science research data,” *Journal of Open Source Software* **8**, 5388 (2023).
- <sup>27</sup>K. H. Kanekal and T. Bereau, “Resolution limit of data-driven coarse-grained models spanning chemical space,” *The Journal of chemical physics* **151** (2019).
- <sup>28</sup>T. Bereau, “Computational compound screening of biomolecules and soft materials by molecular simulations,” *Modelling and Simulation in Materials Science and Engineering* **29**, 023001 (2021).
- <sup>29</sup>P. C. Souza, S. Thallmair, P. Conflitti, C. Ramírez-Palacios, R. Alessandri, S. Raniolo, V. Limongelli, and S. J. Marrink, “Protein–ligand binding with the coarse-grained martini model,” *Nature communications* **11**, 3714 (2020).
- <sup>30</sup>R. Menichetti, K. H. Kanekal, and T. Bereau, “Drug–membrane permeability across chemical space,” *ACS central science* **5**, 290–298 (2019).
- <sup>31</sup>C. Hoffmann, A. Centi, R. Menichetti, and T. Bereau, “Molecular dynamics trajectories for 630 coarse-grained drug-membrane permeations,” *Scientific Data* **7**, 51 (2020).
- <sup>32</sup>A. Centi, A. Dutta, S. H. Parekh, and T. Bereau, “Inserting small molecules across membrane mixtures: Insight from the potential of mean force,” *Biophysical Journal* **118**, 1321–1332 (2020).
- <sup>33</sup>B. Mohr, K. Shmilovich, I. S. Kleinwächter, D. Schneider, A. L. Ferguson, and T. Bereau, “Data-driven discovery of cardiolipin-selective small molecules by computational active learning,” *Chemical Science* **13**, 4498–4511 (2022).
- <sup>34</sup>NOMAD, “NOMAD How-to guides,” <https://nomad-lab.eu/prod/v1/docs/index.html> (), accessed: 2024-08-06.
- <sup>35</sup>marshmallow, “marshmallow: simplified object serialization,” <https://marshmallow.readthedocs.io/en/stable/index.html>, accessed: 2024-07-22.
- <sup>36</sup>MARTIGNAC, “Martignac: Coarse-grained Martini simulation workflows,” <https://tbereau.github.io/martignac/>, accessed: 2024-09-23.
- <sup>37</sup>Martini, “Martini: General Purpose Coarse-Grained Force Field,” <http://www.cgmartini.nl/index.php/force-field-parameters>, accessed: 2024-08-06.
- <sup>38</sup>L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez, “Packmol: A package for building initial configurations for molecular dynamics simulations,” *Journal of computational chemistry* **30**, 2157–2164 (2009).
- <sup>39</sup>M. R. Shirts and J. D. Chodera, “Statistically optimal analysis of samples from multiple equilibrium states,” *The Journal of chemical physics* **129** (2008).
- <sup>40</sup>O. Beckstein, D. L. Dotson, Z. Wu, D. Wille, D. Marson, I. Kennedy, shuail, H. Lee, trje3733, V. Lim, A. Schlaich, I. Alibay, J. Hénin, M. S. Barhaghi, P. Merz, T. Joseph, W.-T. Hsu, helmut carter, and hl2500, “alchemy/alchemyb: 2.3.1,” (2024).
- <sup>41</sup>T. A. Wassenaar, H. I. Ingólfsson, R. A. Bockmann, D. P. Tielemans, and S. J. Marrink, “Computational lipidomics with insane: a versatile tool for generating custom membranes for molecular simulations,” *Journal of chemical theory and computation* **11**, 2144–2155 (2015).
- <sup>42</sup>G. M. Torrie and J. P. Valleau, “Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling,” *Journal of computational physics* **23**, 187–199 (1977).
- <sup>43</sup>S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman, “The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method,” *Journal of computational chemistry* **13**, 1011–1021 (1992).
- <sup>44</sup>M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, “Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers,” *SoftwareX* **1**, 19–25 (2015).
- <sup>45</sup>G. Bussi, D. Donadio, and M. Parrinello, “Canonical sampling through velocity rescaling,” *The Journal of chemical physics* **126** (2007).
- <sup>46</sup>M. Parrinello and A. Rahman, “Polymorphic transitions in single crystals: A new molecular dynamics method,” *Journal of Applied physics* **52**, 7182–7190 (1981).
- <sup>47</sup>Tristan Bereau, “Martignac Streamlit app,” <https://martignac.streamlit.app/>, accessed: 2024-08-22.
- <sup>48</sup>T. D. Potter, E. L. Barrett, and M. A. Miller, “Automated coarse-grained mapping algorithm for the martini force field and benchmarks for membrane–water partitioning,” *Journal of Chemical Theory and Computation* **17**, 5777–5791 (2021).
- <sup>49</sup>S. J. Marrink, A. H. de Vries, and A. E. Mark, “Coarse grained model for semiquantitative lipid simulations,” *The Journal of Physical Chemistry B* **108**, 750–760 (2003).
- <sup>50</sup>D. H. de Jong, G. Singh, W. F. D. Bennett, C. Arnarez, T. A. Wassenaar, L. V. Schäfer, X. Periole, D. P. Tielemans, and S. J. Marrink, “Improved parameters for the martini coarse-grained protein force field,” *Journal of Chemical Theory and Computation* **9**, 687–697 (2012).
- <sup>51</sup>D. Bochicchio, E. Panizon, R. Ferrando, L. Monticelli, and G. Rossi, “Calculating the free energy of transfer of small solutes into a model lipid membrane: Comparison between metadynamics and umbrella sampling,” *The Journal of chemical physics* **143** (2015).