

STOCK: Structure Mapper and Online Coarse-Graining Kit for Molecular Simulations

Staš Bevc,^{*[a]} Christoph Junghans,^[b] and Matej Praprotnik^[a]

We present a web toolkit SStructure mapper and Online Coarse-graining Kit for setting up coarse-grained molecular simulations. The kit consists of two tools: structure mapping and Boltzmann inversion tools. The aim of the first tool is to define a molecular mapping from high, for example, all-atom, to low, that is, coarse-grained, resolution. Using a graphical user interface it generates input files, which are compatible with standard coarse-graining packages, for example, Versatile Object-oriented Toolkit for Coarse-graining Applications and DL_CGMAP. Our second tool generates effective potentials for coarse-grained simulations preserving the structural properties, for example, radial distribution functions, of the underlying

higher resolution model. The required distribution functions can be provided by any simulation package. Simulations are performed on a local machine and only the distributions are uploaded to the server. The applicability of the toolkit is validated by mapping atomistic pentane and polyalanine molecules to a coarse-grained representation. Effective potentials are derived for systems of TIP3P (transferable intermolecular potential 3 point) water molecules and salt solution. The presented coarse-graining web toolkit is available at <http://stock.cmm.ki.si>. © 2014 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23806

Introduction

Coarse-grained molecular simulations are a well established tool for investigating phenomena at larger length- and longer timescales than feasible with an all-atom description of the system.^[1] With a coarser definition of molecules, the number of degrees of freedom are reduced and only the ones essential are retained for the investigation,^[2–10] thus freeing up valuable computational resources. Examples applications include polymers, for example, Refs. [11,12], proteins, for example, Ref. [13], lipid membranes, for example, Refs. [14,15], DNA, for example, Ref. [16], so forth. Coarse-grained models also play an important role in multiscale simulations.^[2] For example, in concurrent type of multiscale simulations different levels of resolution are coupled simultaneously, such that the higher resolution is used for the area of interest and the lower one for the rest of the system, typically the solvent.^[17,18]

Due to various applications of coarse-grained simulations, several coarse-graining methods,^[19–23] tools^[24–31] and simulation packages^[32–34] have been developed over the last couple of years. Some of these tools use their own simulation engines, some rely on third-party simulation software. The tools try to make common methods and simulation techniques appealing and easy to use to a broad scientific community. Nevertheless, these tools typically use configuration files, which first need to be written by an experienced user and second, even so, they require time and effort for application to any new system under study.

In this work, we present a web toolkit using the latest technologies to provide standard coarse-graining tools and methods for setting up coarse-grained molecular simulations. The aim of the presented SStructure mapper and Online Coarse-graining Kit (STOCK) is to facilitate the configuration and preparation of coarse-grained simulations. Using a graphical

user interface it generates input files, which users can download and use on their local machines. The tool is made up of two parts: Structure mapping tool and Boltzmann inversion tool. The structure mapping tool is used to setup coarse-grained topology and weights mapping input files for Versatile Object-oriented Toolkit for Coarse-graining Applications^[26] (VOTCA) or DL_CGMAP^[29] packages. The Boltzmann inversion tool is used to prepare the effective bonded or nonbonded tabulated potentials using iterative Boltzmann inversion.^[20] With iterative Boltzmann inversion, one runs a short all-atom simulation to obtain the required distributions for deriving initial effective potentials for the coarse-grained model and then performs coarse-grained simulations using corrected effective potentials at each iteration until convergence is reached (i.e., distributions match within their fluctuations).^[20] One may use any simulation package or even in-house developed packages with the iterative Boltzmann inversion tool. This is especially useful if one's simulation package of choice is not supported by VOTCA or DL_CGMAP. Currently, VOTCA natively supports GROMACS,^[35] has great support for DL_POLY,^[36] good support for LAMMPS^[37] and basic support for ESPResSo^[38] and ESPResSo++.^[39] DL_CGMAP supports DL_POLY molecular

[a] Staš Bevc, M. Praprotnik

Laboratory for Molecular Modeling, National Institute of Chemistry, Hajdrihova 19, Ljubljana, SI-1001, Slovenia
E-mail: stas.bevc@cmm.ki.si

[b] C. Junghans

Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico, 87545

Contract grant sponsor: Slovenian Research Agency (S.B., M.P.); Contract grant number: J1-4134 and P1-0002; Contract grant sponsor: LANL is operated by Los Alamos National Security, LLC, for the National Nuclear Security Administration of the U.S. DOE; Contract grant number: DE-AC52-06NA25396

© 2014 Wiley Periodicals, Inc.

simulation package. In certain cases, it is not even possible or desired to install additional software in one's environment. Since our tool is web-based, no installation procedure is required.^[40] One just needs a web browser with Internet access. It is freely available at: <http://stock.cmm.ki.si>.

The article is organized as follows. In the next section, we describe our toolkit and the methods used. In the results section, we provide some examples to demonstrate the usefulness of the toolkit and followed by conclusions.

Methods

The STOCK toolkit is made up of two tools, which may be used in conjunction with other coarse-graining packages, but may also be used on their own. Following is a detailed description of both parts of the tool.

Structure mapping tool

The structure mapping tool allows one to map an atomistic representation of a single molecule to a coarse-grained representation. In this context, mapping means grouping together atoms in a coarse-grained bead. Atoms can even be shared between beads. The end result is a topology and weights mapping XML file as is used in VOTCA or DL_CGMAP coarse-graining package. Following is an example of a topology file excerpt, where bead A1 is defined, a bond between beads A1 and A2, and a weights map A, which is used in bead A1 (ellipses denote omission of text):

```
...
<cg_bead>
  <name>A1</name>
  <type>A</type>
  <mapping>A</mapping>
  <beads>1:PNT:C 1:PNT:H 1:PNT:H 1:PNT:H</
  beads>
</cg_bead>
...
<bond>
  <name>bond</name>
  <beads>
    A1 A2
  ...
</beads>
</bond>
...
<map>
  <name>A</name>
  <weights>12 1 1 1</weights>
</map>
...
```

The file is written in XML format and uniquely defines the topology of the coarse-grained molecule. XML is a documents format that is both human-readable and machine-readable. For each molecule type, a new file has to be created. It is composed of several sections enclosed by corresponding start and end tags (e.g., <cg_beads>...</cg_beads>). In between the tags

come definitions of a set of rules. Sections define the molecule name, weight mappings, coarse-grained beads, bonds, angles, and dihedrals. The beads are defined by listing the weights map and atoms that belong to it. Bonds, angles, and dihedrals are defined by listing the beads that form them. Weights are defined by listing the atomic masses separated by spaces.

The position of the coarse-grained site is calculated out of the weights in the following way:

$$\vec{r}_{CG} = \sum_i w_i \vec{r}_i / \sum_i w_i, \quad (1)$$

where the sum goes over all atoms, which are mapped to current coarse-grained site. $w_i = 1$ will lead to a geometrical, while $w_i = m_i$ to a center of mass mapping. One $w_i < 0$ corresponds to a mapping out of the convex hull of the atomistic particles. It is worth noting that noncenter of mass mappings will lead to the wrong distribution in momentum space,^[41] which is why STOCK creates center of mass mappings by default.

Manually writing the topology is prone to errors and can be tedious and cumbersome. Also, one needs to be familiar with the structure and syntax of the XML file. If an error is present in the code, it might be difficult to track down. To overcome this difficulty, we developed a web tool with which one easily creates the input file even for complex molecules using an interactive graphical user interface.

The interface of the structure mapping tool is divided into two main parts: the molecule visualizer and the controls below it (see Fig. 1). The visualizer displays a 3D representation of the molecule and allows one to rotate, translate or zoom the molecule. The controls are divided into Molecule, Weights, Beads, Bonds, Angles, Dihedrals, and XML sections. Sections are organized into tabs. Clicking on tabs switches sections.

After one uploads or selects a molecule file, it is displayed in the visualizer. Any file format supported by Jmol may be used.^[42] One defines beads by switching to the Beads section and then simply selects atoms belonging to a bead. The name and type are predefined for user convenience, but may be changed if desired. The number in the name of the bead is automatically increased when a bead is added. Weight mappings are automatically generated and selected. If one wishes to define them manually, this can be done in the Weights section. When adding beads, one defines the bonds by selecting the corresponding two beads in the drop down menus. The currently selected bond is displayed in the molecular visualizer in yellow. The already added bonds are blue. Beads defining a bond are preselected and automatically advanced when a bond is added. In the most usual case, one just clicks the Add button until all desired bonds are added. The same approach is used for the definition of angles and dihedrals (see Fig. 2). The tool prevents one from adding duplicate or bogus entries. Definition of intramolecular interactions like bonds and angles are additional to the coarse-grained representation, but this information can be used for instance by the VOTCA package to generate exclusions. This can also guide the user about which interactions need to be determined by the Boltzmann inversion tools later on.

STOCK: Structure mapping tool
structure mapper and online coarse-graining kit

Home About

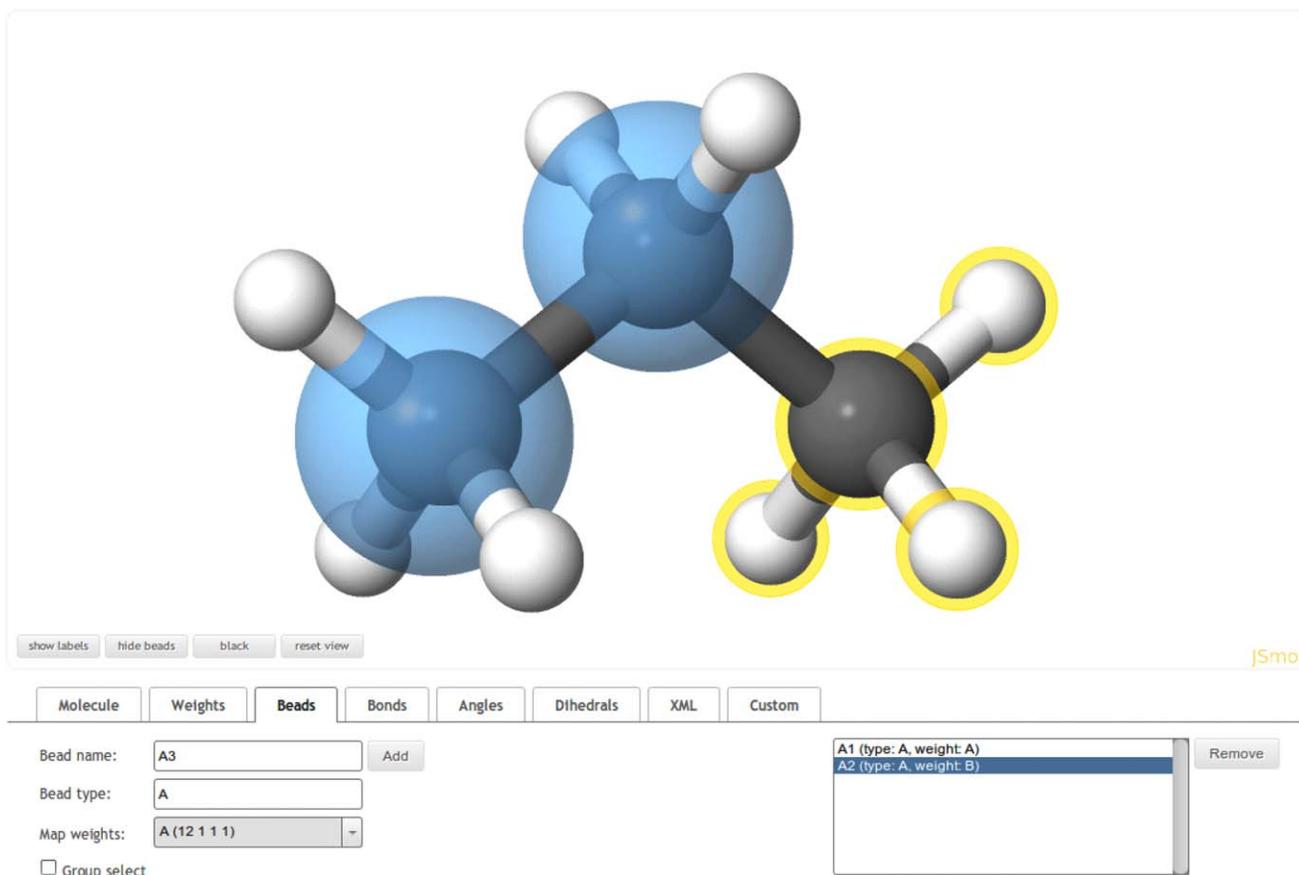


Figure 1. Structure mapping tool. Mapping a propane molecule to a three bead coarse-grained molecule. The beads are displayed in blue. Atoms surrounded by the yellow circles are selected and about to be mapped to a bead.

Each time one selects the XML section, XML code is generated, and defines the current configuration. The code may be downloaded as a file or copied into one's clipboard which can then be pasted where desired. The generated file is readily used as the input topology file for VOTCA or DL_CGMAP package on one's local machine. One may also jump to the Boltzmann inversion tool, by clicking the Switch to Boltzmann button if one does not wish to use VOTCA or DL_CGMAP. The tool prepares a list of all bonded and nonbonded potentials that have to be calculated and are intended to guide the user through the process of coarse graining the corresponding effective potentials.

For the molecule visualizer, we use the recently created JSmol^[43,44] which is a JavaScript implementation of the well established Jmol molecular^[45,46] visualizer.

Boltzmann inversion tool

The Boltzmann inversion tool provides a simple and straightforward way to calculate effective potentials for coarse-grained simulations using the iterative Boltzmann inversion method. In short, this method allows to determine the effective

interaction between coarse-grained particles by inverting the Boltzmann distribution iteratively using the differences of potentials of mean force. The tool may be used with any simulation package, either molecular dynamics (MD) or Monte-Carlo code. Hence, simulations are not run on the server, but instead on the user's machine. One only uploads the distributions from simulations and the tool generates the next iteration of the potential. Since the user is part of the iteration process, more control is on the user's side. Going back a step or two, is easy to do. This prevents scenarios where countless hours on a cluster are wasted if a parameter of the simulation is wrong, but the iterations keep going on. The drawback is, that attention is necessary after every iteration. Convergence is usually reached in around 15 iterations. One is of course free to run as many iterations as desired. Users desiring a fully automatic process are advocated to use other software (e.g., VOTCA), but even in this case, STOCK is useful in the initial stage.

We actively decided to not implement our own MD algorithms, following the same approach as DL_CGMAP and VOTCA, as it is hard, if not impossible, to keep up with the development speed of standard MD packages and to provide

STOCK: Structure mapping tool

structure mapper and online coarse-graining kit

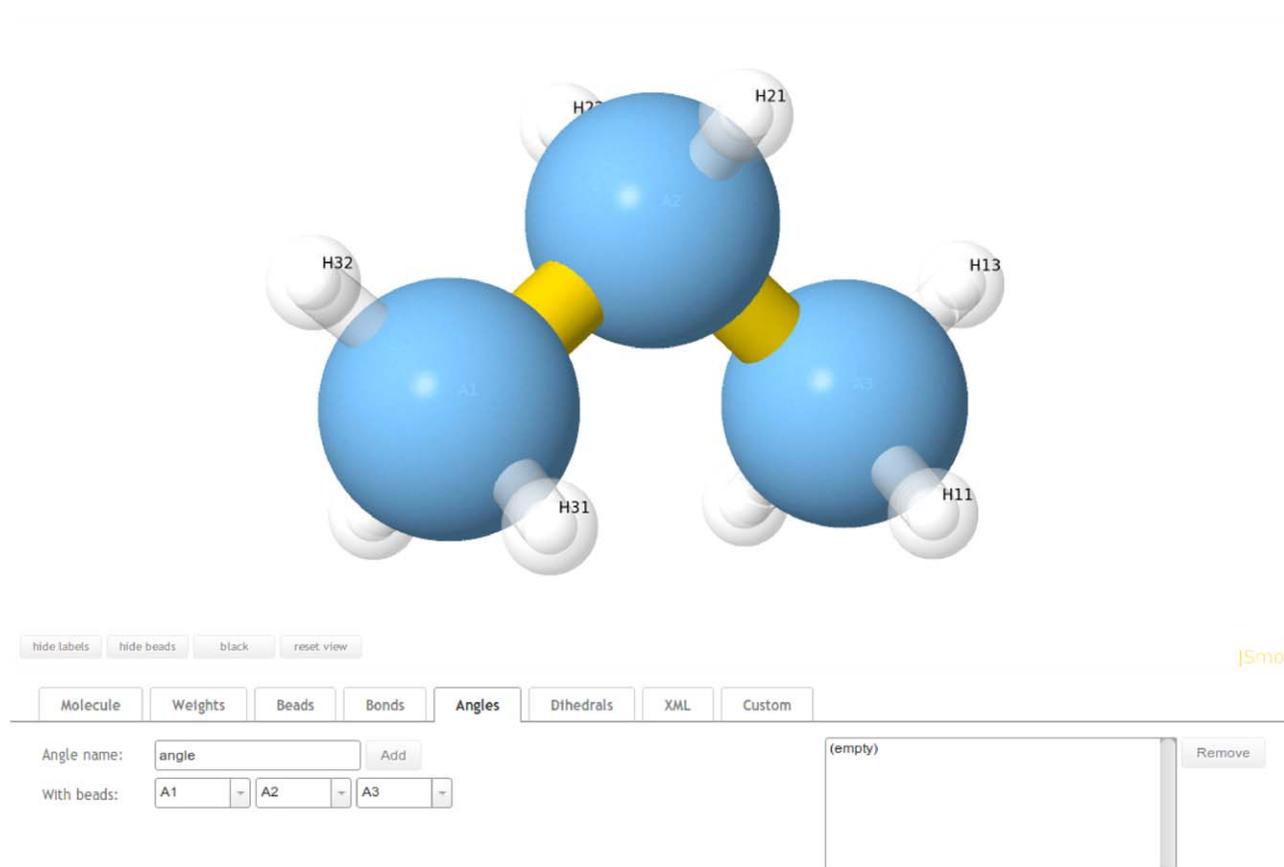
[Home](#) [About](#)


Figure 2. Adding angles to a propane molecule. A triple of beads is selected from the drop down menus (A1, A2, and A3 in this case). The current angle is drawn in yellow. Although adding angles (or bonds/dihedrals) the atoms are translucent. Beads may also be hidden by pressing the hide beads button. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

a comparable feature set. It would also be problematic to provide enough computational power for molecular simulations for every visitor. The same approach is also carried out by other MD web interfaces, for example, CHARMM-GUI^[47] or OpenMM's^[48] script builder which prepare the necessary input files that users then run on their local machine.

As in the structure mapping tool, the user interface is divided into two parts: two charts, and controls below them (as can be seen in Fig. 3). The chart on the left ($P(r)$) displays distributions of the target atomistic system and the obtained distributions from the coarse-grained simulations. The chart on the right ($U(r)$) displays the calculated potentials from a particular distribution. Colors of the distributions in the left chart, generally match with the color of the corresponding potential in the right chart. The charts are interactive and can be zoomed and panned using a mouse. A vertical ruler follows one's mouse and is synchronized between the two charts such that the x-position of the ruler is the same on both charts. This makes it easy to determine if a bump in the distribution is caused by a bump in the potential and vice-versa. Ranges of x- and y-axis may be set using the x_r and y_r input boxes on the corresponding chart.

Controls are comprised of three input select boxes displaying the uploaded or generated files, corresponding to target distribution files, obtained distribution files and potential files. Four buttons are below each input box: the plus button (+) adds a file from one's local machine, the minus button (−) removes a selected file, the Smoother button smooths a selected file and the Download button which downloads the selected file. The gear button allows one to choose the format of potential files. On the right side are parameters for the potential generation (range, cutoff, grid density, the $k_B T$ constant, and pressure correction parameters) and a Generate potential button. Below the controls is a status window which displays the history of the user's or system's actions. The colors of the file names in the panels and in the status window also match the colors in the charts.

The usage procedure is remarkably simple: first one (step 1) uploads the target distribution files (one or more) in the Target distribution input box. This distribution is usually calculated from an atomistic simulation and serves as a reference target for the coarse-grained simulation. As soon as a file is uploaded, the initial potential guess file is automatically generated in the Potential input box. Both files are displayed in their

STOCK: Boltzmann inversion tool
structure mapper and online coarse-graining kit

Home About

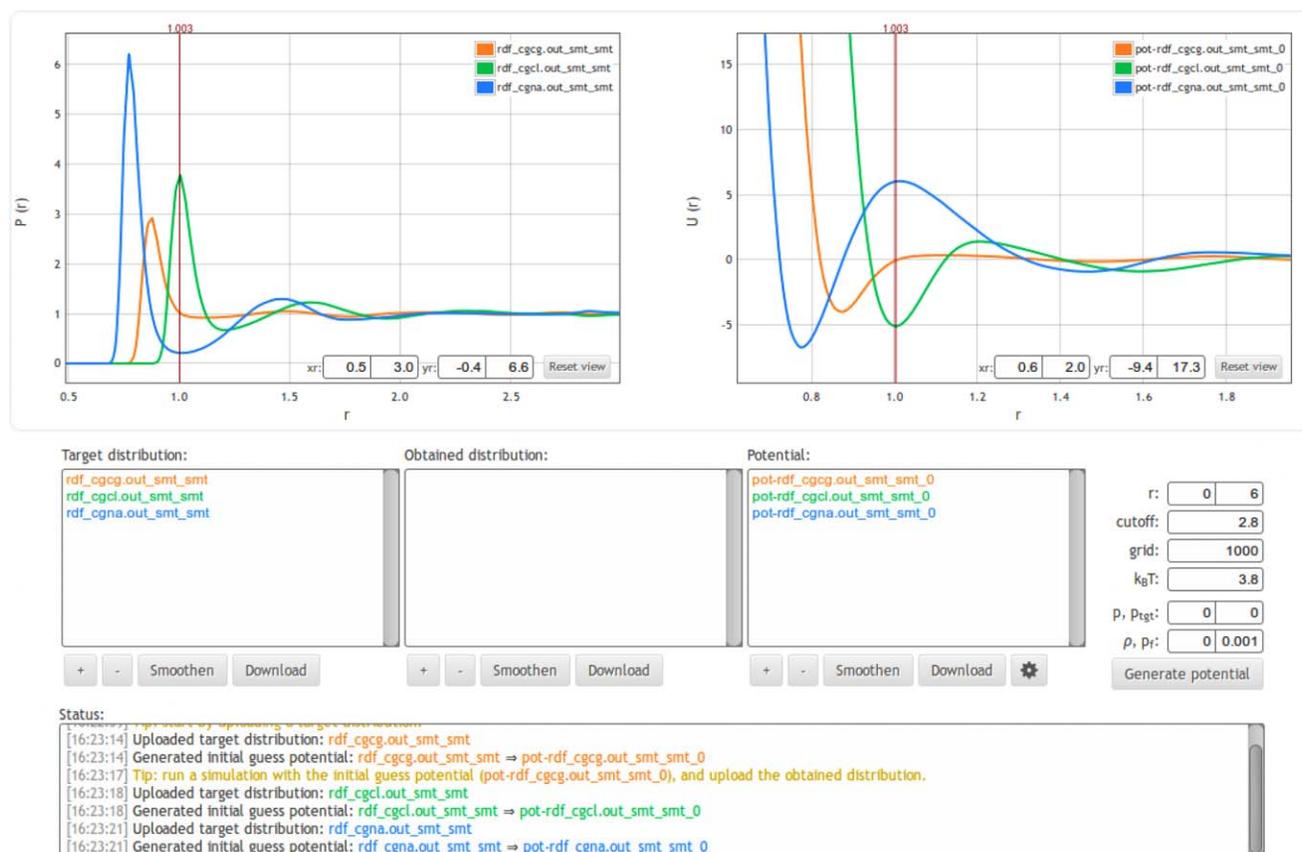


Figure 3. Boltzmann inversion tool. In the above figure, RDFs and initial guess potentials are shown between center of mass water molecules (orange curves), and between center of mass water molecules and ions (green and blue curves) for a salt solution system. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

corresponding charts. One (step 2) downloads the generated potential files and runs a short coarse-grained simulation using them. The obtained distribution functions from this simulation are then (step 3) uploaded in the Obtained distribution input box. After that, one selects (step 4) the corresponding three files in the target, obtained and potential panels, and presses the Generate potential button (i.e., one connects the potential with the obtained distribution and the desired target distribution). A new potential is generated in the Potential input box. One then iterates steps 2, 3, and 4 until the desired target distribution is obtained. Usage procedure is illustrated in Figure 4.

Smoothing of data is supported. On pressing the Smoothen button, a window opens displaying the original and the smoothed curve. Smoothing is applied in real time which allows one to visually inspect the outcome and tune the smoothing parameters until reaching the best results. Since smoothing may smear out steep peaks, we made it possible to do smoothing in two separate ranges independently. This is useful to smooth the region before and after the first peak with different parameters, but ranges are allowed to overlap. Data points which are not covered by the smoothing range

remain unchanged. If the ranges overlap, smoothing in first range is applied first and then smoothing from the second range is added. The potential is automatically smoothed at the beginning when creating the initial guess, and afterwards smoothing is only applied to the potential update, and not the total potential anymore, to avoid the above mentioned smearing out of peaks. But the user may apply additional smoothing if wanted or needed. Smoothing is implemented with convolution of a Kaiser window over the data. It is also possible to do interpolation of data points with various splines in two ranges. One is of course able to smooth an already smoothed file, for even finer control.

The initial potential guess U_0 is calculated as the potential of the mean force:

$$U_0(r) = -k_B T \ln P_{\text{tgt}}(r), \quad (2)$$

where k_B is the Boltzmann constant, T is the temperature and $P_{\text{tgt}}(r)$ is the distribution function of a target system. After one runs a coarse-grained simulation with this potential, the obtained distribution will most likely not match the target distribution. The potential thus needs to be refined, which is done iteratively. The next iteration of the potential U_{i+1} is calculated as:

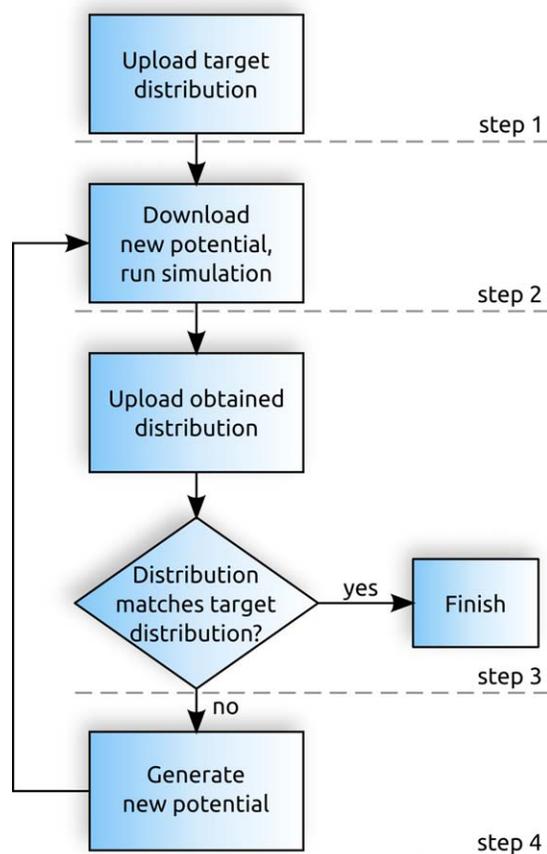


Figure 4. Usage diagram for the Boltzmann inversion tool. Steps 2, 3, and 4 are iterated until the coarse-grained distribution matches the target distribution. Simulation in step 2 is run locally on one's machine. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$U_{i+1}(r) = U_i(r) + k_B T \ln \left[\frac{P_i(r)}{P_{\text{tgt}}(r)} \right]. \quad (3)$$

The process is repeated until convergence is reached, that is, the distribution function $P_i(r)$ matches the target distribution function $P_{\text{tgt}}(r)$. For bonded interactions, the initial potential needs to be normalized differently, see Ref. [26,49].

Since iterative Boltzmann inversion reproduces structural properties of the underlying atomistic system but not the thermodynamic properties, the pressure of the coarse-grained system might not match the pressure of the reference atomistic system. To match the pressures, an additional correction can be applied to the effective potential. A simple linear function:

$$\Delta U_i(r) = A_i \left(1 - \frac{r}{r_{\text{cut}}} \right) \quad (4)$$

is added to the coarse-grained potential as suggested in Ref. [20], where A_i is calculated as:

$$A_i = -\text{sgn}(p_i - p_{\text{tgt}}) 0.1 k_B T_{\text{min}} (1, |f(p_i - p_{\text{tgt}})|) \quad (5)$$

(as implemented in VOTCA 1.1^[50]. p_i is the current system

pressure, and p_{tgt} is the target pressure, f is a scaling factor. Simple pressure correction is applied if one enters the current and target pressure of the system. If one enters the particle density ρ as well, pressure correction as proposed in Ref. [51] will be applied instead. In this case, A_i is calculated from the virial expansion:

$$A_i = (p_i - p_{\text{tgt}}) / \left[\frac{-2\pi\rho^2}{3r_{\text{cut}}} \int_0^{r_{\text{cut}}} r^3 g_i(r) dr \right] \quad (6)$$

Calculations of potentials are done on the server and the results are sent to one's browser where they are displayed. The format of distribution files is two columns separated by spaces or tabs. The first column is r and the second is $P(r)$. The format of potential files may be chosen among several popular formats, and also providing a generic format, where the first column is r and the second column is $U(r)$. When force $F(r)$ is provided, it is calculated as the negative derivative of potential with respect to r . Force is not used in the iteration process, it is calculated for one's convenience only. All units are arbitrary but must be consistent.

Every web page visitor of the Boltzmann inversion tool is assigned a random string which is used as the identification of the project. One may bookmark the web page and return to it at anytime. This also allows one to send a link to a particular project or to resume an existing project from a different location or computer.

Results and Discussion

The applicability and functionality of the presented toolkit is illustrated on four examples. For the first two examples, we use the structure mapping tool, and for the second two examples, we use the Boltzmann inversion tool. The examples are also available at the STOCK webpage <http://stock.cmm.ki.si>.

Example 1: Mapping a pentane molecule

Here, we create a topology file for a coarse-grained pentane molecule. The molecule is composed of 17 atoms and we map it to a coarse-grained representation of the molecule composed of five beads (see Fig. 5). After uploading the file in the Molecule section, we enter the names of the atomistic and coarse-grained representations. The weights are set to be automatically mapped by default, so we do not enter them. We proceed to the Beads section, where we click the first carbon atom and its connecting hydrogens. The bead name and bead type are left to their default values. We add the bead, and repeat the process for the remaining four carbon atoms. Next, we add the bonds, angles and dihedrals just by pressing the Add button in their corresponding sections. On switching to the XML section, the following topology input file is generated:

```

<cg_molecule>
  <name>PNT</name>
  <ident>Pentane</ident>
  <topology>
  <cg_beads>

```

```
<cg_bead>
  <name>A1</name>
  <type>A</type>
  <mapping>A</mapping>
  <beads>1:UNK:C 1:UNK:H 1:UNK:H 1:UNK:H</
beads>
</cg_bead>
<cg_bead>
  <name>A2</name>
  <type>A</type>
  <mapping>B</mapping>
  <beads>1:UNK:C 1:UNK:H 1:UNK:H</beads>
</cg_bead>
<cg_bead>
  <name>A3</name>
  <type>A</type>
  <mapping>B</mapping>
  <beads>1:UNK:C 1:UNK:H 1:UNK:H</beads>
</cg_bead>
<cg_bead>
  <name>A4</name>
  <type>A</type>
  <mapping>B</mapping>
  <beads>1:UNK:C 1:UNK:H 1:UNK:H</beads>
</cg_bead>
<cg_bead>
  <name>A5</name>
  <type>A</type>
  <mapping>A</mapping>
  <beads>1:UNK:C 1:UNK:H 1:UNK:H 1:UNK:H</
beads>
</cg_bead>
</cg_beads>
<cg_bonded>
  <bond>
    <name>bond</name>
    <beads>
      A1 A2
      A2 A3
      A3 A4
      A4 A5
    </beads>
  </bond>
</angle>
  <name>angle</name>
  <beads>
    A1 A2 A3
    A2 A3 A4
    A3 A4 A5
  </beads>
</angle>
<dihedral>
  <name>dihedral</name>
  <beads>
    A1 A2 A3 A4
    A2 A3 A4 A5
```

```
</beads>
  </dihedral>
</cg_bonded>
</topology>
<maps>
  <map>
    <name>A</name>
    <weights>12 1 1 1</weights>
  </map>
  <map>
    <name>B</name>
    <weights>12 1 1</weights>
  </map>
</maps>
</cg_molecule>
```

The file may be downloaded and is ready to be used in VOTCA or DL_CGMAP coarse-graining package. Generating the above XML file takes less than a minute, while writing it by hand would take considerably longer.

Example 2: Mapping a polyaniline molecule

A typical example of large molecules mapped to coarse-grained representations are polymers such as Polystyrene^[52,53] (107 atoms, 13 beads). In this example, we map a polyaniline molecule with composed of 103 atoms to a 51-bead coarse-grained representation (as seen in Fig. 6). As in the previous example, we upload the molecule file, enter the names of the atomistic and coarse-grained representations, and proceed to add the beads. We coarse-grain the molecule such that each heavy atom with its attached hydrogen(s) is mapped to one coarse-grained bead.^[54] After adding the beads, we add bonds, angles, and dihedrals between neighboring beads, as we described before. When switching to the XML section, 548 lines are generated (not shown here) with definitions for 51 beads, 50 bonds, 67 angles, 74 dihedrals, and 7 weight mappings. With this example, we have shown that it is possible to define mappings even for such complex molecules as a polyaniline.

Example 3: Coarse-graining water molecules

To assess the Boltzmann inversion tool, we derive coarse-grained water–water nonbonded effective potentials such that the structural properties of the underlying atomistic system are reproduced. The coarse-grained model is set up by mapping a water molecule to a one-bead representation located at the center of mass of the molecule. The simulation box contains 510 TIP3P^[55] water molecules (1530 atoms), with a side length of 2.5 nm. Standard periodic boundary conditions and the minimum image convention are used. Temperature is regulated at 300°K using a local Langevin thermostat with the value of friction constant 5 ps⁻¹. All simulations are run for 1 ns with the ESPResSo++ simulation package.

We first run an atomistic simulation and calculate the radial distribution function (RDF) between centers of mass of water molecules. We upload the RDF as the target distribution in

STOCK: Structure mapping tool

structure mapper and online coarse-graining kit

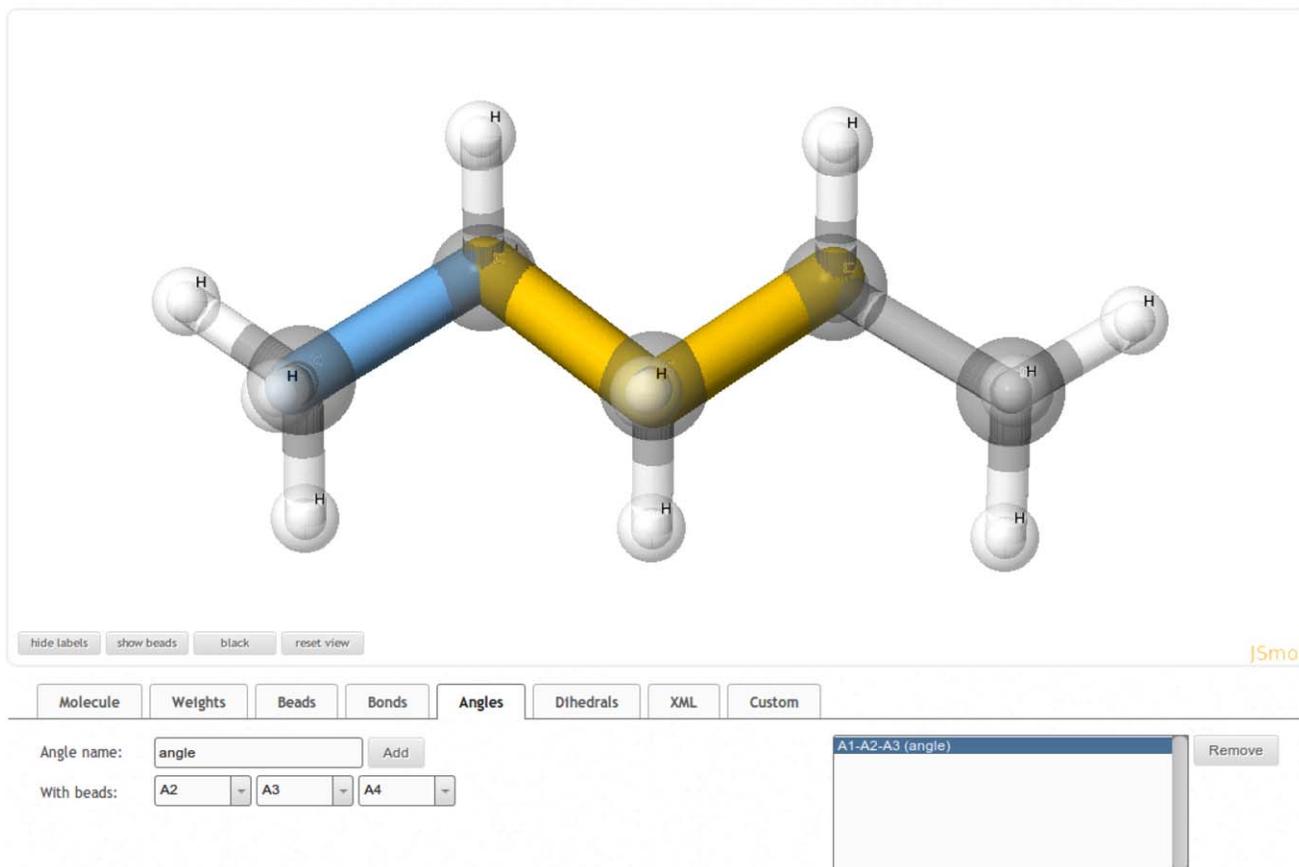
[Home](#) [About](#)


Figure 5. Adding an angle to a pentane molecule. Beads are hidden in the above figure. This is accomplished by pressing the hide beads button. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

STOCK (see Fig. 7) and smoothen it from the point $r = 0.33$ nm onwards. We interpolate the curve with an Akima^[56] spline to get a nice smooth curve. Next, we generate the initial guess potential using our smoothed target distribution file. Range is set from 0 to 2 nm, cutoff is at 0.9 nm, grid is 1000, and $k_B T$ is 3.8. No pressure correction is applied. These same parameters are kept for all iterations.

With the initial potential guess, we run a coarse-grained simulation, calculate the RDF and upload it as the obtained distribution. We smoothen it in a similar way as the target distribution, making sure not to impair the first peak. By connecting the target distribution, the obtained distribution and the used potential, we generate the next iteration of the potential and use it to run a new coarse-grained simulation. We repeat these steps until reaching a satisfied matching (shown in Fig. 8) between our obtained RDF and the target RDF after 29 iterations.

Example 4: Coarse-graining a salt solution

In this example, we derive effective potentials for a NaCl salt solution, that is, between water–water, water–Na, and water–Cl. The coarse-grained model is set up by mapping a water

molecule to a one-bead representation located at the center of mass of molecule. Atomistic simulation is run to obtain reference RDFs. The simulation setup for the all-atom system is the same as in Ref. [18]. We use a standard SPC/E^[57] water model with AMBER^[58] forcefield for ions. Generalized reaction field method^[59] is used for electrostatic interactions with cut-off radius $R_c = 0.9$ nm, $\epsilon_1 = 1$, $\epsilon_2 = 80$, and the Debye screening length $\kappa = 3.25$ nm⁻¹. Integration is done with standard velocity Verlet with a time-step of 1 fs. A Langevin thermostat with the friction constant 15 ps⁻¹ is used to regulate temperature at 300°K. The size of the system is $16.08 \times 4.02 \times 4.02$ nm³, containing 8377 water molecules and 157 Na and 157 Cl ions, corresponding to a 1 M salt solution. The geometry of water molecules is constrained with SETTLE^[60] algorithm. All simulations are run for 1 ns with ESPResSo++ simulation package.

Having calculated the atomistic RDFs (water–water, water–Cl, water–Na), we upload them in STOCK as the target distributions. We smoothen them in a similar way as described in the previous example and generate the three initial guess potentials. Parameters for the generated potentials are the same as

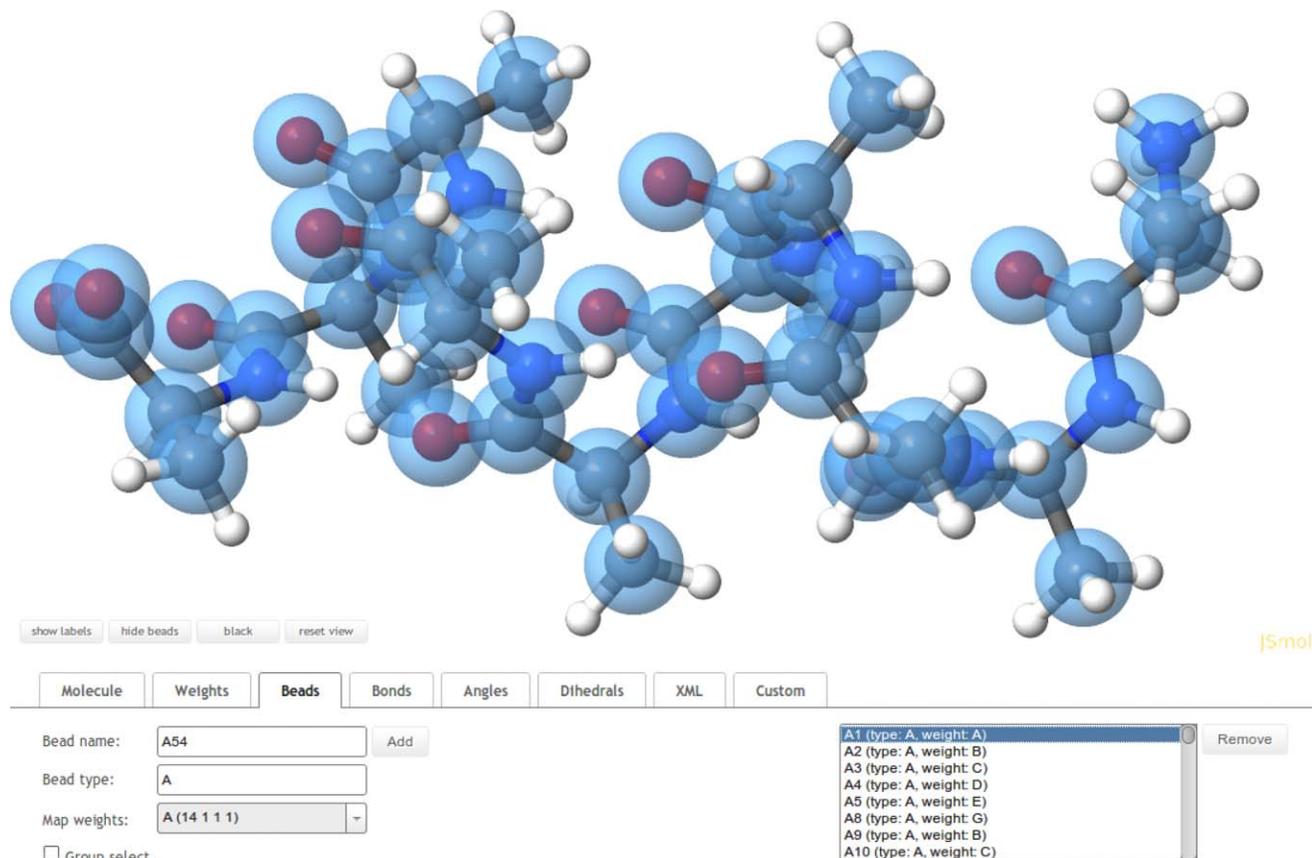
STOCK: Structure mapping tool
structure mapper and online coarse-graining kit[Home](#) [About](#)

Figure 6. Mapping a polyaniline molecule. Heavy atoms with its attached hydrogens are mapped to one coarse-grained bead. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

in the previous example, and are kept for all iterations. Pressure correction is not applied.

We run a coarse-grained simulation using these potentials for water–water and water–ions interactions. For ion–ion interactions, Lennard-Jones parameters are the same as in the atomistic simulation, but for the generalized reaction field we set $\epsilon_1 = 80$ to properly screen the electrostatic interactions between ions.

After conducting the coarse-grained simulation, we calculate the RDFs and upload them as the obtained distributions. We smooth them, then connect the target distribution, the obtained distribution and the potential for water–water and generate the next iteration of the potential. Next, we connect the two distributions and potential for water–Cl to generate the next iteration of the potential, and lastly we do it in the same manner for water–Na. Thus, in every iteration we create three new potentials and run a new coarse-grained simulation using them. We repeat the process until reaching convergence, that is, the RDFs of center of mass of water–water, water–Cl, and water–Na match the atomistic RDFs. Resulting potentials and RDFs are shown in Figure 9 and were calculated in 17 iterations. The results computed by STOCK are in good agreement

with the RDFs and effective potentials from Ref. [18] obtained by VOTCA.

Conclusions

In this article, we presented a toolkit for setting up coarse-grained molecular simulations. Our web structure mapper greatly facilitates the coarse-grained mapping setup and is compatible with popular coarse-graining packages, for example, VOTCA and DL_CGMAP. The Boltzmann inversion tool for the effective potentials derivation, as the second tool of the STOCK web toolkit, may be used with any simulation package. The structure mapper tool offers a point-and-click way to map a high resolution representation of a molecule to a lower resolution with the aid of a molecular visualizer and outputs an XML mapping, which represents the standard for mapping files these days. The Boltzmann inversion tool calculates effective potentials for coarse-grained simulations, such that the structural properties of the underlying higher resolution system are preserved. Simulations are not run on the server, which makes the tool software independent. One may use it with their simulation package of choice. Our mapping tool is most suitable

STOCK: Boltzmann inversion tool

structure mapper and online coarse-graining kit

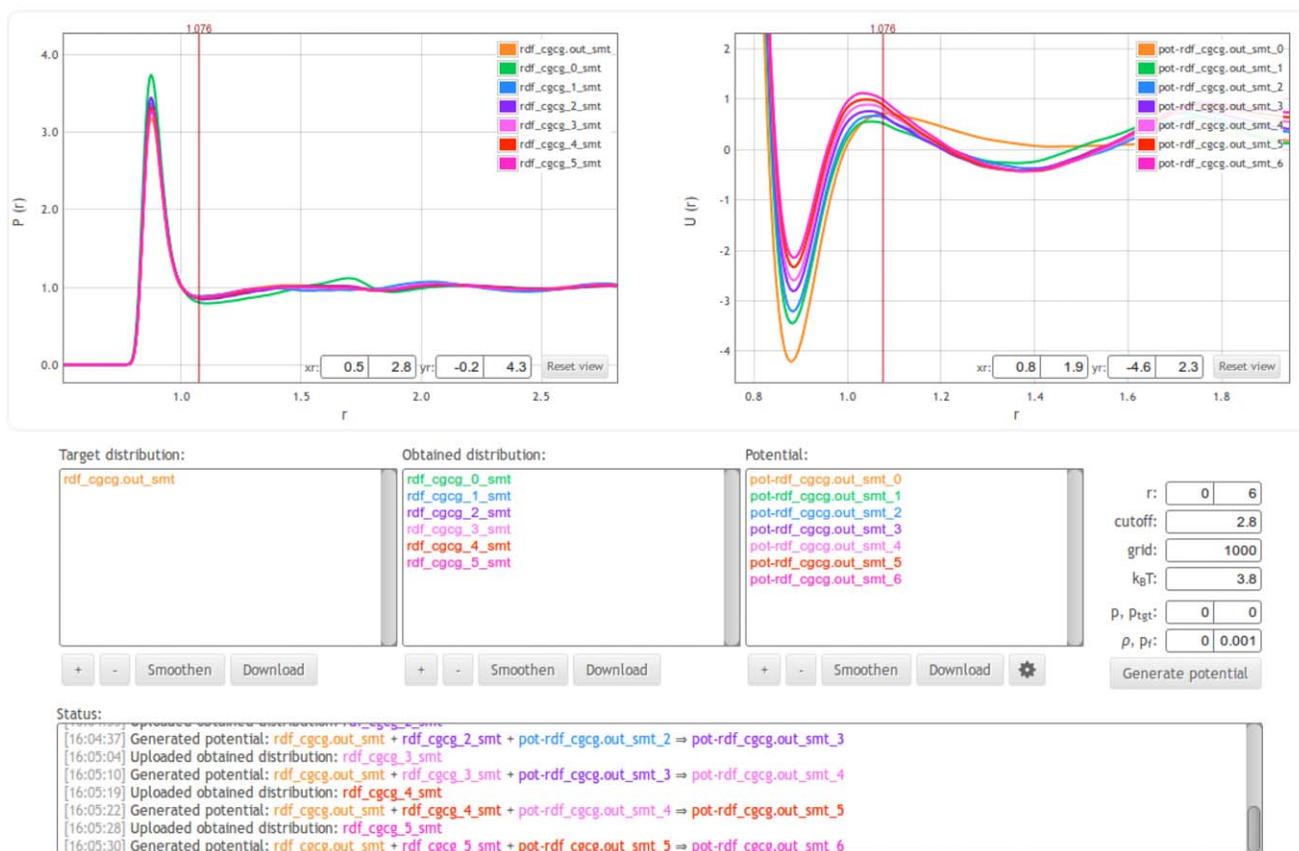
[Home](#) [About](#)


Figure 7. Coarse-graining water molecules. The figure shows part of the process of creating effective potentials for TIP3P water molecules. Reduced units are used in the figure. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

for molecules of moderate sizes (of the order of 100 atoms). However, if one wants to use STOCK with larger molecules, such as proteins or nucleic acids, it requires more user input. For example, very large molecule can be split into smaller parts, for protein that would be residues. We demonstrated

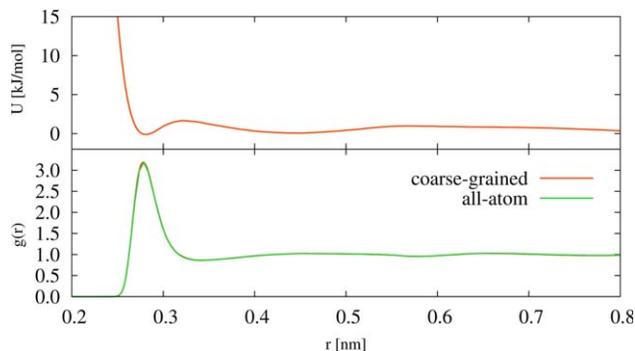


Figure 8. Calculated nonbonded effective potential for water-water is shown on top. Radial distribution functions for an all-atom and coarse-grained simulation are shown at the bottom. RDFs match very well. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the applicability of the structure mapping tool by mapping atomistic pentane and polyaniline molecules to a coarse-grained representation. The Boltzmann inversion tool is

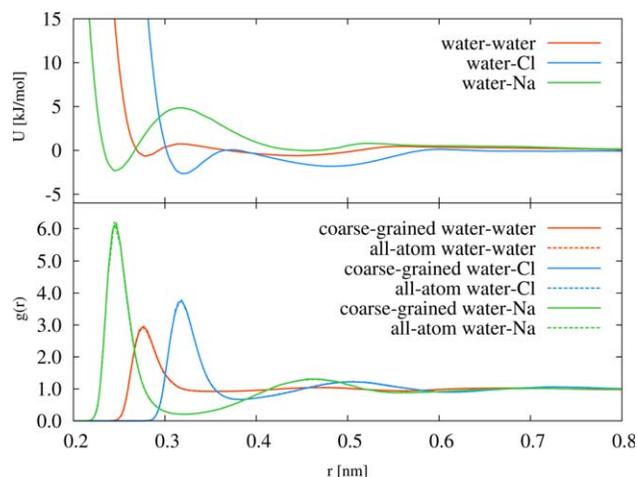


Figure 9. Effective pair potentials for water-water, water-Cl, and water-Na are shown on top. Radial distribution functions for an all-atom and coarse-grained simulation are shown at the bottom. Atomistic RDFs are very well reproduced by the coarse-grained simulation.

assessed by coarse-graining systems of TIP3P water molecules and NaCl salt solution. We leverage on latest web technologies to provide a simple to use toolkit, that we believe will lower the entrance barrier to researchers wanting to dive into coarse-grained simulations.

Acknowledgments

The authors thank Sue Mniszewski for critical reading of the manuscript. S. B. and M. P. acknowledge financial support through the grants J1-4134 and P1-0002 from the Slovenian Research Agency. C. J. thanks Los Alamos National Laboratory for a Director's Post-doctoral fellowship. Assigned: LA-UR-14-21684.

Keywords: coarse-graining · molecular simulations · iterative Boltzmann · structure mapping · web tool

How to cite this article: S. Bevc, C. Junghans, M. Praprotnik. *J. Comput. Chem.* **2014**, *36*, 467–477. DOI: 10.1002/jcc.23806

- [1] S. C. Kamerlin, S. Vicatos, A. Dryga, A. Warshel, *Annu. Rev. Phys. Chem.* **2011**, *62*, 41.
- [2] M. Praprotnik, L. Delle Site, K. Kremer, *Annu. Rev. Phys. Chem.* **2008**, *59*, 545.
- [3] M. Praprotnik, S. Pobleto, K. Kremer, *J. Stat. Phys.* **2011**, *145*, 946.
- [4] C. Peter, L. Delle Site, K. Kremer, *Soft Matter* **2008**, *4*, 859.
- [5] S. Izvekov, G. A. Voth, *J. Chem. Phys.* **2005**, *123*, 134105.
- [6] G. S. Ayton, W. G. Noid, G. A. Voth, *Curr. Opin. Struct. Biol.* **2007**, *17*, 192.
- [7] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, A. H. de Vries, *J. Phys. Chem. B* **2007**, *111*, 7812.
- [8] A. Chaimovich, M. S. Shell, *Phys. Rev. E* **2010**, *81*, 060104.
- [9] Z.-J. Wang, M. Deserno, *J. Phys. Chem. B* **2010**, *114*, 11207.
- [10] W. G. Noid, *J. Chem. Phys.* **2013**, *139*, 090901.
- [11] B. Hess, S. Leon, N. van der Vegt, K. Kremer, *Soft Matter* **2006**, *2*, 409.
- [12] V. A. Harmandaris, K. Kremer, *Macromolecules* **2009**, *42*, 791.
- [13] A. Davtyan, N. P. Schaefer, W. Zheng, C. Clementi, P. G. Wolynes, G. A. Papoian, *J. Phys. Chem. B* **2012**, *116*, 8494.
- [14] A. J. Rzepiela, M. Louhivuori, C. Peter, S. J. Marrink, *Phys. Chem. Chem. Phys.* **2011**, *13*, 10437.
- [15] B. J. Reynwar, G. Illya, V. A. Harmandaris, M. M. Muller, K. Kremer, M. Deserno, *Nature* **2007**, *447*, 461.
- [16] J. J. de Pablo, *Annu. Rev. Phys. Chem.* **2011**, *62*, 555.
- [17] M. Praprotnik, L. Delle Site, K. Kremer, *J. Chem. Phys.* **2005**, *123*, 224106.
- [18] S. Bevc, C. Junghans, K. Kremer, M. Praprotnik, *New J. Phys.* **2013**, *15*, 105007.
- [19] W. Tschöp, K. Kremer, J. Batoulis, T. Bürger, O. Hahn, *Acta Polym.* **1998**, *49*, 61.
- [20] D. Reith, M. Pütz, F. Müller-Plathe, *J. Comput. Chem.* **2003**, *24*, 1624.
- [21] A. P. Lyubartsev, A. Laaksonen, *Phys. Rev. E* **1995**, *52*, 3730.
- [22] A. Soper, *Chem. Phys.* **1996**, *202*, 295.
- [23] S. Izvekov, G. A. Voth, *J. Chem. Phys.* **2005**, *123*, 134105.
- [24] CGTools plugin for VMD, Available at: <http://www.ks.uiuc.edu/Research/vmd/plugins/cgtools/>. Accessed 21st October, 2014.
- [25] D. Reith, H. Meyer, F. Müller-Plathe, *Comput. Phys. Commun.* **2002**, *148*, 299.
- [26] V. Rühle, C. Junghans, A. Lukyanov, K. Kremer, D. Andrienko, *J. Chem. Theory Comput.* **2009**, *5*, 3211.
- [27] H. A. Karimi-Varzaneh, H.-J. Qian, X. Chen, P. Carbone, F. Müller-Plathe, *J. Comput. Chem.* **2011**, *32*, 1475.
- [28] A. Mirzoev, A. P. Lyubartsev, *J. Chem. Theory Comput.* **2013**, *9*, 1512.
- [29] CCP5 New Flagship Project: Coarse Grain Modelling, Available at: http://www.ccp5.ac.uk/projects/ccp5_cg.shtml. Accessed 21st October, 2014.
- [30] T. A. Wassenaar, K. Pluhackova, R. A. Böckmann, S. J. Marrink, D. P. Tieleman, *J. Chem. Theory Comput.* **2014**, *10*, 676690.
- [31] Y. Qi, X. Cheng, W. Han, S. Jo, K. Schulten, W. Im, *J. Chem. Inf. Model.* **2014**, *54*, 1003.
- [32] T. Aoyagi, F. Sawa, T. Shoji, H. Fukunaga, J. Takimoto, M. Doi, *Comput. Phys. Commun.* **2002**, *145*, 267.
- [33] R. K. Z. Tan, A. S. Petrov, S. C. Harvey, *J. Chem. Theory Comput.* **2006**, *2*, 529.
- [34] A. Górecki, M. Szymowski, M. Długosz, J. Trylska, *J. Comput. Chem.* **2009**, *30*, 2364.
- [35] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4*, 435.
- [36] W. Smith, I. T. Todorov, *Mol. Simul.* **2006**, *32*, 935.
- [37] S. Plimpton, *J. Comput. Phys.* **1995**, *117*, 1.
- [38] H. J. Limbach, A. Arnold, B. A. Mann, C. Holm, *Comput. Phys. Commun.* **2006**, *174*, 707.
- [39] J. D. Halverson, T. Brandes, O. Lenz, A. Arnold, S. Bevc, V. Starchenko, K. Kremer, T. Stuehn, D. Reith, *Comput. Phys. Commun.* **2013**, *184*, 1129.
- [40] S. Bevc, J. Konc, J. Stojan, M. Hodošček, M. Penca, M. Praprotnik, D. Janežič, *PLoS ONE* **2011**, *6*, e22265.
- [41] W. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, H. C. Andersen, *J. Chem. Phys.* **2008**, *128*, 244114.
- [42] Jmol: list of supported file formats, Available at: http://wiki.jmol.org/index.php/File_formats/Coordinates. Accessed 21st October, 2014.
- [43] R. M. Hanson, J. Prilusky, Z. Renjian, T. Nakane, J. L. Sussman, *Isr. J. Chem.* **2013**, *53*, 207.
- [44] JavaScript-based molecular viewer from Jmol, Available at: <http://sourceforge.net/projects/jsmol/>. Accessed 21st October, 2014.
- [45] M. E. Cass, H. S. Rzepa, D. R. Rzepa, C. K. Williams, *J. Chem. Educ.* **2005**, *82*, 1736.
- [46] Jmol: an open-source Java viewer for chemical structures in 3D, Available at: <http://www.jmol.org/>. Accessed 21st October, 2014.
- [47] S. Jo, T. Kim, V. G. Iyer, W. Im, *J. Comput. Chem.* **2008**, *29*, 1859.
- [48] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, V. S. Pande, *J. Chem. Theory Comput.* **2013**, *9*, 461.
- [49] C. Peter, K. Kremer, In *Multiscale Simulation Methods in Molecular Sciences*, Vol. 42 of NIC Series; J. Grotendorst, N. Attig, S. Blügel, D. Marx, Eds.; Forschungszentrum Jülich: Jülich, **2009**, pp. 337–358.
- [50] V. Rühle, C. Junghans, *Macromol. Theory Simul.* **2011**, *20*, 472.
- [51] H. Wang, C. Junghans, K. Kremer, *Eur. Phys. J. E* **2009**, *28*, 221.
- [52] D. Fritz, V. A. Harmandaris, K. Kremer, N. F. A. van der Vegt, *Macromolecules* **2009**, *42*, 7579.
- [53] D. Fritz, K. Koschke, V. A. Harmandaris, N. F. A. van der Vegt, K. Kremer, *Phys. Chem. Chem. Phys.* **2011**, *13*, 10412.
- [54] W. Han, Y.-D. Wu, *J. Chem. Theory Comput.* **2007**, *3*, 2146.
- [55] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, *J. Chem. Phys.* **1983**, *79*, 926.
- [56] H. Akima, *J. ACM* **1970**, *17*, 589.
- [57] H. J. C. Berendsen, J. R. Grigera, T. P. Straatsma, *J. Phys. Chem.* **1987**, *91*, 6269.
- [58] J. Wang, P. Cieplak, P. A. Kollman, *J. Comput. Chem.* **2000**, *21*, 1049.
- [59] I. G. Tironi, R. Sperb, P. E. Smith, W. F. van Gunsteren, *J. Chem. Phys.* **1995**, *102*, 5451.
- [60] S. Miyamoto, P. A. Kollman, *J. Comput. Chem.* **1992**, *13*, 952.

Received: 7 August 2014

Revised: 14 November 2014

Accepted: 19 November 2014

Published online on 15 December 2014