Martini tutorials: running Martini simulations with ddcMD
(if you just landed here, we recommend you follow the tutorials about lipid bilayers and proteins first)

Intro:
The Martini force field has been implemented within Lawrence Livermore National Laboratory's molecular dynamics program, ddcMD. This implementation has been fully ported to GPU. By executing all computations on the GPU ddcMD-Martini frees up CPU resources for other tasks and allows for full GPU utilization on systems with low CPU to GPU ratio. For a description of this implementation and benchmarking see Zhang et al. 2020 [**1**].

ddcMD-Martini has been very usefully in previous [**2,3,4**] and current MuMMI campaigns, allowing for good utilization of all GPUs (6 per node on *Summit* and 4 per node on *Sierra* and *Lassen*) and allowed us to use the available CPUs for running of out macro model, setup and initial equilibration of simulations and online analysis of running simulations.

Warning, currently virtual-sites are not supported in ddcMD-Martini; adding virtual-sites is currently being worked on and expected shortly. This means that Martini 3 proteins with virtual site amino acids will not work and for Martini 2 non-virtual-sites versions molecules will have to be used. Cholesterol is the most common one and a version of the original non-virtual-site Martini cholesterol (v01) but with updates to the cholesterol shape to reflect that of the never-virtual-sites model (v02) can be found here https://bbs.llnl.gov/RAS-lipid-dependent-dynamics-data.html called (martini_v2.0_CHOL_01-fix.itp).

Install:
First, we will download and install ddcMD, a GROMACS to ddcMD converter and MDAnalysis update to read ddcMD format.
ddcMD - https://github.com/LLNL/ddcMD
ddcMD converter - https://github.com/LLNL/ddcMDconverter
updated MDAnalysis - https://github.com/XiaohuaZhangLLNL/mdanalysis

Install ddcMD - see https://github.com/LLNL/ddcMD/blob/master/INSTALL.md

```
git clone git@github.com:LLNL/ddcMD.git
# git clone https://github.com/LLNL/ddcMD.git
cd ddcMD
git submodule update --init --recursive # Get dependent libraries
mkdir build && cd build
module load gcc/7.3.1  # Add gcc to path
module load cuda/10.1.243  # Add CUDAA compiler to path
cmake -DUSE_GPU=ON -DCMAKE_INSTALL_PREFIX:PATH=<location of binary> ../
make -j16
make install
```

Install ddcMD compatible version of MDAnalysis:

```
git clone git@github.com:XiaohuaZhangLLNL/mdanalysis.git
# git clone https://github.com/XiaohuaZhangLLNL/mdanalysis.git
cd mdanalysis/package
pip install -e .
```

Install ddcMDconverter, converts between GROMACS and ddcMD formats:

```
git clone https://github.com/LLNL/ddcMDconverter.git
# git clone https://github.com/LLNL/ddcMDconverter.git
cd ddcMDconverter
pip install -e .
```

Simulation setup and equilibration:

We suggest you run energy minimization and initial system equilibration using GROMACS and after system is ready for production run convert the GROMACS files to ddcMD input.
An example pre-equilibrated simulation is given (ddcMD-Martini-tautorial.tar.gz). This is a 3200-lipid membrane of an 8-component plasma membrane mimic [**5**] with one RAS peripheral membrane protein attached to the inner leaflet, a typical patch from the first MuMMI campaign [**3**, input data].

Convert GROMACS to ddcMD format:

Use martini2obj (part of ddcMDconverter) to convert GROMACS parameters/itp files to ddcMD parameter files. Outputted files are martini.data, molecule.data, ConsAtom.data; which contain all the simulation parameters.

All the GROMACS itp files and associated input files are in the para directory. Note the parser is fairly simple so the itp files need to "cleaner" than what GROMACS allows: no special characters, no if/def statements etc.

```
cp -r inputs/para .
cd para
martini2obj -i itpList -t proItpList -p martini_v2.1-dna.itp -o martini.data
-l molecule.data
cd ../
```

Copy the martini.data molecule.data ConsAtom.data to ddcmd simulation directory

```
mkdir sim
cp para/martini.data para/molecule.data para/ConsAtom.data sim/
cd sim
```

Use pdbmartini2obj (part of ddcMDconverter) to convert .pdb file and GROMACS .top file to the ddcMD coordinates files: atoms#000000 and the restart files.

```
cp ../inputs/gromacs/* .
pdbmartini2obj -p lipids-water-eq4.pdb -t system.top -m martini.data -f
ConsAtom.data -o atoms#000000
```

If the simulation contains position restraints, they need to be specified separately. In this simulation the upper leaflet contains POPC lipids (here called POPX) with weak contains on the

PO4 bead z-axes only to limit bilayer undulations. Use restraint (part of ddcMDconverter) to generate the restraint.data (for POPX) require in this simulation (resItpList specifies the GROMACS input files with constraints and the -i parameter the reference restraint values).

```
cp ../inputs/para/resItpList .
cp ../inputs/para/POPX_Martini_v2.0_lipid.itp .
restraint -i atoms#000000 -o restraint.data -p resItpList
```

Now setup the simulation restart location and copy in the default simulation run option file (object.data is similar to a GROMACS .mdp file the main options are described in the file)

```
mkdir -p snapshot.mem
mv atoms#000000 restart snapshot.mem
ln -s snapshot.mem/restart # Set start at initial frame
cp ../inputs/para/object.data .
```

Run with ddcMD:

Make sure the following inputs are presented:

```
molecule.data
martini.data
object.data
restraint.data
restart # symbolic link file
snapshot.mem
    atoms#000000
    restart
```

Run the simulation

```
<your ddcMD binary> -o object.data molecule.data
```

Simulation analysis:

Using the modified version of MDAnalysis installed above ddcMD coordinate files (both full atom restart files and smaller binary files) can be opened, new files can be appended to open MDAnalysis universe objects and combined tar ddcMD trajectory files can be read in.

Open single ddcMD snapshot

```
python
>> import MDAnalysis as mda
>> u = mda.Universe("topol.tpr", "snapshot.mem/atoms#000000", format='DDCMD')
```

New ddcMD snapshots can be

```
>> u.load_new(frameName, format='DDCMD')
```

Open ddcMD trajectory (tar file containing multiple snapshots). Note remember to adjust the timestep (dt) and in MDAnalysis dt is in ps and current time step of ddcMD is set in the object.data. Using MDAnalysis a ddcMD trajectory can then be saved down to other MD formats, such as GROMACS .xtc

```
for frame in `seq -w 1000 1000 000000010000 `
do
    cp snapshot.${frame}/"subset#000000" pos.${frame}
```

```
done
tar -cvf positions.tar pos.000*
>> u = mda.Universe("topol.tpr", "positions.tar", format='DDCMDTAR', dt=20.0)
>> with mda.Writer("ddcMD_trj.xtc", u.trajectory.n_atoms) as w:
>>     for ts in u.trajectory:
>>         w.write(u.atoms)
```

References:

[1] X Zhang, S Sundram, T Oppelstrup, SIL Kokkila-Schumacher, TS Carpenter, HI Ingólfsson, FH Streitz, F C Lightstone, JN Glosli. 2020. ddcMD: A fully GPU-accelerated molecular dynamics program for the Martini force field. *J. Chem. Phys.* 153:045103. doi: 10.1063/5.0014500.

[2] F Di Natale, H Bhatia, TS Carpenter, C Neale, SIL Kokkila-Schumacher, T Oppelstrup, L Stanton, X Zhang, S Sundram, TRW Scogland, G Dharuman, MP Surh, Y Yang, C Misale, L Schneidenbach, C Costa, C Kim, B D'Amora, S Gnanakaran, DV Nissley, F Streitz, FC Lightstone, PT Bremer, JN Glosli, HI Ingólfsson. 2019. A Massively Parallel Infrastructure for Adaptive Multiscale Simulations: Modeling RAS Initiation Pathway for Cancer. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '19). ACM, New York, NY, USA, 57:16. doi: 10.1145/3295500.3356197.

[3] HI Ingólfsson, C Neale, TS Carpenter, R Shrestha, CA López, TH Tran, T Oppelstrup, H Bhatia, LG Stanton, X Zhang, S Sundram, F Di Natale, A Agarwal, G Dharuman, SIL Kokkila-Schumacher, T Turbyville, G Gulten, QN Van, D Goswami, F Jean-Francios, C Agamasu, D Chen, JJ Hettige, T Travers, S Sarkar, MP Surh, Y Yang, A Moody, S Liu, BC Van Essen, AF Voter, A Ramanathan, NW Hengartner, DK Simanshu, AG Stephen, PT Bremer, S Gnanakaran, JN Glosli, FC Lightstone, F McCormick, DV Nissley, FH Streitz. 2020. Machine Learning-driven Multiscale Modeling Reveals Lipid-Dependent Dynamics of RAS Signaling Proteins. doi: 10.21203/rs.3.rs-50842/v1 (Preprint).

[4] H Bhatia, F Di Natale, JY Moon, X Zhang, JR Chavez, F Aydin, C Stanley, T Oppelstrup, C Neale, S Kokkila-Schumacher, DH Ahn, S Herbein, TS Carpenter, S Gnanakaran, PT Bremer, JN Glosli, FC Lightstone, HI Ingólfsson. 2021. Generalizable Coordination of Large Multiscale Workflows: Challenges and Learnings at Scale. In The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '21), November 14–19, 2021, St. Louis, MO, USA. ACM, New York, NY, USA, 14. doi: 10.1145/3458817.3476210.

[5] HI Ingólfsson, H Bhatia, T Zeppelin, WFD Bennett, KA Carpenter, PC Hsu, G Dharuman, PT Bremer, B Schiøtt, FC Lightstone, TS Carpenter. 2020. Capturing biologically complex tissue-specific membranes at different levels of compositional complexity, *J. Phys. Chem. B*, 124:7819–7829. doi: 10.1021/acs.jpcb.0c03368.

[**Note**] Tutorial written by Helgi I. Ingólfsson, Xiaohua Zhang and Jim Glosli. LLNL-MI-825738.